

SEPTEMBER 1988

COMPUTER



TITAN GRAPHICS SUPERCOMPUTER

Also in this issue:



Soviet High-Speed Computers
Fuzzy Concepts in Expert Systems
Strategic Management of R&D

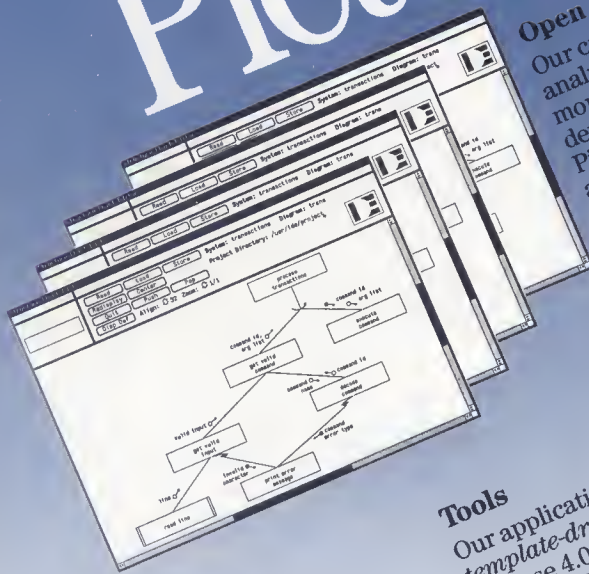
 IEEE COMPUTER SOCIETY

 THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

Computer Society Election

Announcing
Release 4.0

Software through Pictures™



Open Architecture

Our customers loved our structured analysis and design tools but wanted more help across the *entire* software development life cycle. Software through Pictures' open architecture provides a foundation that integrates well with other tools and allows you and your development environment to evolve together.

Database

Our *multi-user* database supports your project team even better on a heterogeneous network. Our tools are built on top of a new object-oriented data dictionary with links to Apollo's DSEE, DEC's CMS, and Sun's NSE.

Tools

Our applications use new *template-driven* tools. Release 4.0 features include a Document Preparation System, an Object Annotation Editor, and SQL schema generation.

Applications

Release 4.0 delivers the most-requested productivity solutions with *user-extensible* CASE applications: Automatic Documentation, Data Dictionary Analysis, Requirements Traceability, and DOD-STD-2167 support.

Software through Pictures is available for the Apollo, DEC VAXstation (VMS and Ultrix), HP 9000, and Sun workstations.

Delivering the Promise of CASE™

Call or Write Today



Interactive Development Environments (IDE)
595 Market Street, 12th Floor
San Francisco, California 94105
415/543-0900

FAX 415/543-3716

Telex 256 697 IDE UR

Electronic mail at sales@ide.com (Internet)
or uunet!ide!sales (uucp)

© 1988 IDE. Software through Pictures is a trademark of Interactive Development Environments, Inc. The other companies mentioned own numerous registered trademarks.

Breakthrough in presentation of simulation results

SIMSCRIPT II.5 with SIMANIMATION

Now you see an animated picture of the system



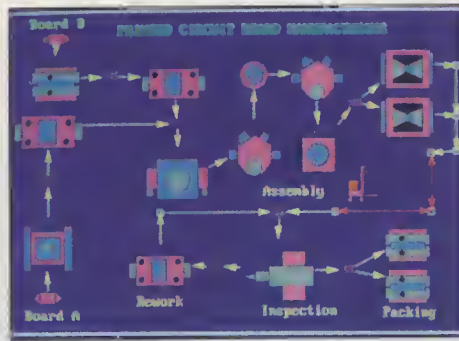
SIMGRAPHICS™ scenario preparation



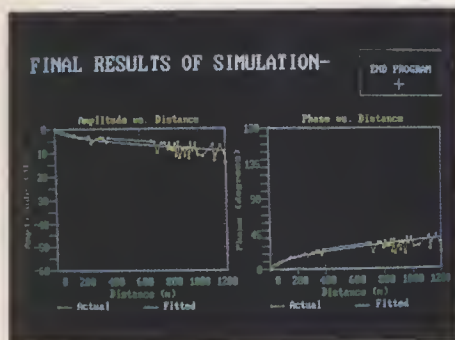
SIMGRAPHICS menu builder



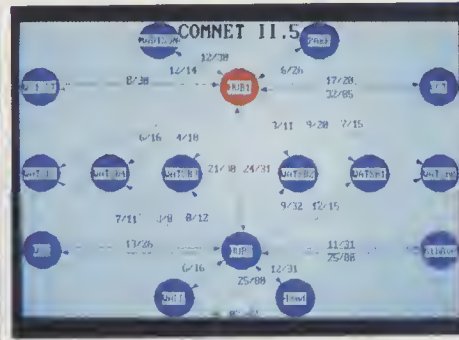
Computer communications-NETWORK II.5®



Factory in operation-SIMFACTORY®



Presentation graphics



Telecommunications-COMNET II.5®



Map with moving objects



Combined discrete-continuous

Free trial and training

See for yourself how simulation results are now easier to understand.

The free trial contains everything you need to try SIMANIMATION® on your computer.

We send you SIMSCRIPT II.5, animated models, and complete documentation. You can build your own model or modify one of ours.

Try the SIMSCRIPT II.5® language, the timeliness of our support, the accuracy of our documentation, and the facilities for error checking--everything you need for a successful project.

No cost, no obligation.
Act now for free training

For a limited time we will also include free training.

For immediate information

Call Hal Duncan at (619) 457-9681. In the UK, call Richard Eve on (01) 528-7980. In Canada call, Steve Hazan on (416) 941-9310.

Rush information on SIMSCRIPT II.5 with SIMANIMATION

Free trial—learn the reasons for the broad and growing popularity of SIMSCRIPT II.5 with SIMANIMATION. **No cost, no obligation.**

Limited offer—return the coupon today and we will also include one free course enrollment worth \$950.

☐ Send information on your special university offer.

Name _____ Telephone _____

Organization _____

Address _____

City _____ State _____ Zip _____

Computer _____ Operating System _____

Return to: _____ IEEE COMP

CACI Products Company
3344 North Torrey Pines Court
La Jolla, California 92037
Call Hal Duncan at (619) 457-9681

In the UK:
CACI Products Company
Regent House, 89 Kingsway
London, WC2B 6RH, United Kingdom
Call Richard Eve on (01) 528-7980

In Canada:
CACI Products Company
One First Canadian Place, 59th Floor
Toronto, Ontario M5X 1K2, Canada
Call Steve Hazan at (416) 941-9310

SIMSCRIPT II.5, SIMANIMATION, COMNET II.5, NETWORK II.5, and SIMGRAPHICS are registered trademarks and service marks of CACI, INC.
©1988 CACI, INC.

COMPUTER

September 1988

Published by the IEEE Computer Society

Vol. 21, No. 9

ARTICLES

13 The Titan Graphics Supercomputer Architecture

Tom Diede, Carl F. Hagenmaier, Glen S. Miranker, Jonathan J. Rubinstein, and William S. Worley, Jr.

Leading-edge hardware and software technologies now make possible a new class of system—the graphics supercomputer. Titan architecture provides a substantial fraction of supercomputer performance plus integrated high-quality graphics.

32 High-Speed Computers of the Soviet Union

Peter Wolcott and Seymour E. Goodman

In the last decade, Soviet development of high-speed/advanced-architecture computers has significantly increased in scope, profile, and support.

43 Fuzzy Concepts in Expert Systems

K.S. Leung and W. Lam

The expert system shell System Z-11 handles both exact and inexact reasoning. It allows any combination of fuzzy and normal terms and uncertainties.

SPECIAL REPORT

58 National Conference on Strategic Management of R&D

Ware Myers and Marilyn Potes

Decision-makers from industry, government, and academia gathered to examine issues that influence a national agenda for R&D and US competitiveness in the world marketplace.

COMPUTER

10662 Los Vaqueros Circle, Los Alamitos, CA 90720

Editor-in-Chief

Bruce D. Shriver, University of Hawaii

Editorial Board

Dharma P. Agrawal, North Carolina State University

Dennis R. Allison, Stanford University

Scott Davidson, AT&T

Richard Eckhouse, MOCO

Scott E. Fahlman, Carnegie Mellon University

Gideon Frieder, Syracuse University

Edmund Gallizzi, Eckerd College

John Gurd, University of Manchester

James H. Haynes, University of California, Santa Cruz

Wiley McKinzie, Rochester Institute of Technology

K.M. Mohiuddin, IBM Almaden Research Laboratory

Yale Patt, University of California, Berkeley

William Riddle, Software Productivity Consortium

Sallie Sheppard, Texas A&M University

Helen M. Wood, National Bureau of Standards

Akinori Yonezawa, Tokyo Institute of Technology

Submissions

Send six double-spaced copies of articles and special-issue proposals to Bruce D. Shriver, Dept. of Decision Sciences, University of Hawaii, 2404 Maile Way, Honolulu, HI 96822; phone, (808) 948-7231. For electronic submission, Shriver's addresses are b.shriver on Comppmail + , shriver@uhccux on Bitnet, and shriver@uhccux.uhcc.hawaii.edu.

Staff

Editor and Publisher: True Seaborn

Assistant Publisher: Douglas L. Combs

Assistant to the Publisher: Pat Paulsen

Managing Editor: Marilyn Potes

Assistant Editors: Nancy Hays, Chuck Governale, Steve H. Wilcox

Art Director: Jay Simpson

Design/Production: Miriam Wiegel

Membership Manager: Christina Champion

Circulation Manager: Paul Zive

Advertising Director: Dawn Peck

Advertising Coordinators: Heidi Rex, Marian Tibayan

Magazine Advisory Committee

Sallie Sheppard (chair), Sumit DasGupta, Michael Evangelist, James J. Farrell III, Ted Lewis, David Pessel, True Seaborn, Bruce D. Shriver, John Staudhammer

Publications Board

James H. Aylor (chair), Victor Basili, J. Richard Burke, Jon T. Butler, J.T. Cain, Sumit DasGupta, Michael Evangelist, James J. Farrell III, Glen G. Langdon, Ted Lewis, Ming T. Liu, Theo Pavlidis, David Pessel, Sallie Sheppard, Bruce D. Shriver, Charles B. Silio, John Staudhammer, Harold Stone, Steven L. Tanimoto

DEPARTMENTS



6 President's Message

Society's financial condition "excellent"

68 Standards

Supercomputer benchmarking

72 Computer Society News

Kyoto Prize; Robotics TC; Section 1706; Compusat; copyright ruling; education report; Computer Entrepreneur Award

77 Computer Society Election

Candidates' position statements and biographies

88 Update

90 New Product Reviews

Ada on the IBM PC

98 New Products

104 IC Announcements

106 Microsystem Announcements

112 Conferences

114 Calendar

120 Call for Papers

123 Book Reviews

Programming Windows; MAP/TOP Networking; 68000 Microcomputer Systems; Compact Disc-Interactive; Law and the Software Marketer

127 New Literature

Computer Society Information Page, inside back cover; Membership Application, 119; Student Membership Application, 105; Change-of-Address Form, 111; Career Opportunities, 108; Advertiser/Product Index, 128; Reader Service Card, 128A

Cover image: In its classical form, a Möbius strip — named after August Möbius (1790-1868), a German mathematician — is the one-sided surface formed when one end of a rectangular strip is twisted one half turn and attached to the opposite end. Variations and generalizations can be obtained by varying the number of twists and increasing the width of the strip to introduce self-intersections.

The Möbius strip surface shown on our cover was produced on Ardent Computer's Titan Graphics Supercomputer using Matlab and DORE software. Matlab (Matrix Laboratory) is an interactive matrix and mathematical visualization package developed by MathWorks, Sherborn, Massachusetts. DORE (Dynamic Object Rendering Environment) is Ardent's graphics and visualization toolkit.

For additional images rendered on a Titan, see p. 16.

Cover design: Jay Simpson, Design & Direction

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society: IEEE Headquarters, 345 East 47th St., New York, NY 10017; Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; phone, (714) 821-8380. Annual subscription included in society member dues. Nonmember prices: available upon request. Single copy prices: members \$10.00; nonmembers \$20.00. This magazine is also available in microfiche form.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854. Second class postage is paid at New York, New York, and at additional mailing offices.

Copyright and reprint permission: Copyright © 1988 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons: (1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress Street, Salem, MA 01970; (2) pre-1978 articles without fee. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint, or republication permission, write to True Seaborn, Editor and Publisher, *Computer*, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space considerations. *Computer* subscribes to the Computer Press Association's code of professional ethics.

BPA

National Design

CAN THIS BE DONE?



A COMMUNICATION
DEVICE FOR THE
SPEECH - DEAF

EXECUTIVE AUDIAL
REHABILITATION
SOCIETY. . . .

the EARS design competition

FIRST PRIZE - \$100,000
PLUS A % OF ROYALTY

SECOND PRIZE - \$50,000
PLUS A % OF ROYALTY

THIRD PRIZE - \$25,000
PLUS A % OF ROYALTY

EXECUTIVE AUDIAL REHABILITATION SOCIETY

500 AMERICAN BANK TOWER P. O. Box 2703 CORPUS CHRISTI, TEXAS 78475
(512)882-6557

JUDGES COMMITTEE

JACK S. KILBY, Dallas, Texas, Electronics Engineer
HOWARD H. ARNOLD, Lewisville, North Carolina, Electronics Engineer
JAMES S. NEWCOMB, St. Paul, Minnesota, Electronics Engineer
HUBERT W. UPTON, Arlington, Texas, Electronics Engineer — Alternate
PROFESSOR HUGH TOMLINSON, Corpus Christi, Texas — Coordinator

WINNERS 1986 SUGGESTION CONTEST

MILDRED CRUICKSHANK, Telephone connection
ED POTTHARST, Multichannel memory

Competition

WE THINK YOU CAN!

THERE ARE 17,000,000 speech-deaf persons, i.e. do not understand speech, in the U.S. The objective is to provide a communication device that will allow one-on-one communication. The final EARS design, incorporating all possible operational advantages and economies of production, may be the composite of the prize winners' designs and may also incorporate features of the designs submitted by non-prize winners. Winners will be accorded cash and royalties. **All royalties negotiated by EARS will be passed through to the contestants.** The royalties will be apportioned by the Judges on the basis of their evaluation of the importance of each element submitted by any contestant which is included in the Final Production Design.

Those desiring to participate should apply (by mail or phone) for a registered entry blank to the Executive Audial Rehabilitation Society shown. Each entry shall include design, programming instruction and an operating "bread board" model which need not conform to the size goals. EARS will compensate those contestants (whose entries are adjudged to be operationally feasible) for the current cost of the components used to build the bread board operational model. The entries will be evaluated by the Judges on the basis of merit, i.e. operationally effective that can be produced at a reasonable cost.

Detailed conditions of the competition will be set out on the entry form. DEADLINES: Entry forms must be procured and returned to EARS not later than October 31, 1988. Designs and bread board models must be received by 5 P.M. on May 1, 1989.

All designs submitted and accepted by the Judges shall be the property of EARS. No essential conceptual element of a design shall constitute a conflict with an existing patent. Conceptual protection of the device has been established as of date of origin. Formal application for patents on pertinent features will be applied for by EARS. A firm of patent attorneys specializing in electronics is retained. All decisions by the Judges as to the award or non-award of prizes or royalty shall be final and shall not be subject to change, review or amendment by EARS., its Officers, Executive Committee or the design contestants.

DESIGN GOALS

1 The goal is a design for a hand-held, lightweight real time communication device capable of being produced at economical cost. The design should allow a speaking person to communicate on a one-on-one real time basis with the hearing-impaired person in any situation where normal conversation occurs and will provide for filtration of extraneous noise. The goal is a device contained in approximately 6" x 8" x 1½". A 5" x 7" screen will be on one side. The speech-deaf person can place it on a desk or hold it in hand. The speaker will first read a text identifying his phonetic alphabet in the memory. A multi-position switch would allow storage of the phonetic alphabets of different persons. For telephone calls, the speech of a person (whose phonetic alphabet is in one of the memory channels or who reads the text into a new channel) will enter the communication device through a standard telephone plug. The device will be battery-powered with rechargeable batteries but provide for alternative operation with a voltage converter. Intensive electronic speech analysis combined with the personal phonetic alphabet of the speaker will result in the "real-time" production of the spoken words on the screen. The speech of the hearing-impaired person not fitting the phonetic alphabet of the memory should be filtered out automatically if possible, otherwise manually.

2 Equally important is a design for this device that can be manufactured economically so that the cost is within the means of this group of people, many of whom may be economically disadvantaged.

Society's financial condition 'excellent'



Edward A. Parrish, Jr.

This month I am pleased to have the society's treasurer, Barry Johnson, detail financial status as of year-end 1987. It represents quite a turnaround from the previous year, thanks to the hard work of both staff and volunteers.

Edward A. Parrish, Jr.
President

A key attribute that allows the Computer Society to fulfill its mission is financial health and vitality. A strong financial position has—to mention just a few examples—allowed the society to expand its publications, increase the number of sponsored conferences, enhance membership recruitment and retention efforts, and provide new funding for standards activities. In addition, the society has opened offices in both Brussels and Tokyo, thereby expanding our international component and making it easier for members to take advantage of our services. I am very pleased to report that the society is in excellent financial condition and is well-positioned to continue to provide—and expand—many outstanding services to the membership.

The accompanying auditor's report from Coopers & Lybrand shows the society's current assets, liabilities, fund balance, revenue, and expenses for 1986 and 1987. Highlights of the report include:

- Surpluses of \$528,200 in 1987 and \$105,800 in 1986.
- Increases of approximately 15 percent and 12 percent in revenue and expenses, respectively, between 1986 and 1987.
- Approximately 86 percent of all expenses in 1987 were associated with publications, conferences, and other technical activities, while only 14 percent went to administration.
- Membership fees accounted for approximately 7.5 percent of total revenue in 1987, while subscriptions and publication activities accounted for approximately 58.5 percent.

A final important point to note is that the society is quickly correcting any problems identified in earlier reports. Specifically, prior years' audited reports



Barry W. Johnson

had included a qualification based on unavailability of supporting documentation for revenue and expenses of some conferences for which the accounting was not done in the society's own offices. Computer Society staff have since implemented additional reporting controls and a system of rotating audits of such conferences. This allowed the auditors to remove that qualification for 1987, and it is not expected to be a problem again.

In summary, the financial condition of the Computer Society is excellent. With the support, leadership, and efforts of our membership, volunteer leaders, and staff, the society can look forward to continued technical and financial success.

Barry W. Johnson
Computer Society Treasurer

Auditor's report

Computer Society 1987, 1986 financial statements

Continuing a policy initiated four years ago, Coopers & Lybrand has examined the Computer Society's financial records. The accounting firm's report for the years ended December 31, 1987 and 1986, appears below.

Board of Governors of the
Computer Society of the IEEE:

We have examined the balance sheets of the Computer Society of the IEEE as of December 31, 1987 and 1986, and the

related statements of revenue, expenses, and changes in fund balance for the years then ended. Except as explained in the following paragraph, our examinations were made in accordance with generally accepted auditing standards and, accordingly, included such tests of the accounting records and such other auditing procedures as we considered necessary in the circumstances.

As discussed in Note 1 to the financial statements, the society reports revenue and expenses of all conferences partially or entirely sponsored by the society. However, for certain conferences held in 1986, revenue and expenses were reported solely on the final representations of the external conference administrators. Consequently, the society did not maintain or control supporting information for all conference revenue and expenses. Therefore, we were not able to examine sufficient documentary evidence in 1986 with respect thereto.

In our opinion, except for the effects of such adjustments, if any, as might have been determined to be necessary had we been able to examine sufficient documentary evidence with respect to 1986 conference revenue and expenses, the financial statement referred to above present fairly the financial position of the Computer Society of the IEEE as of December 31, 1987 and 1986, and the results of its operations for the years then ended, in conformity with generally accepted accounting principles applied on a consistent basis.

Coopers & Lybrand
1800 M Street, NW
Washington, DC 20036
April 15, 1988

The Computer Society of the IEEE Balance Sheets December 31, 1987 and 1986

Assets	1987	1986
Current assets:		
Cash, including interest-bearing accounts	\$ 228,300	\$ 558,200
Investments (Note 3)	2,737,100	1,502,100
Accounts receivable, less allowance of \$81,300 in 1987 and \$111,700 in 1986	575,900	670,300
Receivable from Institute of Electrical and Electronics Engineers, Inc. (Note 7)	43,000	71,300
Conference receivables	710,300	423,100
Conference advances	165,800	139,000
Inventory	398,200	420,000
Prepaid expenses and other assets	580,100	697,600
Total current assets	5,438,700	4,481,600
Fixed assets, net (Note 4)	3,658,700	3,393,200
Total assets	\$ 9,097,400	\$ 7,874,800
Liabilities and Fund Balance		
Current liabilities:		
Accounts payable and accrued expenses	\$ 1,162,500	\$ 1,074,800
Notes payable (Note 5)	540,200	33,000
Deferred income:		
Membership fees and subscriptions	3,065,900	2,521,500
Conferences	19,800	127,600
Advertising and other	58,400	118,500
Total current liabilities	4,846,800	3,875,400
Notes payable, less current portion	1,366,100	1,643,100
Total liabilities	6,212,900	5,518,500
Fund balance:		
Undesignated	2,793,100	2,277,000
Designated for technical committees (Note 8)	91,400	79,300
Total fund balance	2,884,500	2,356,300
Total liabilities and fund balance	\$ 9,097,400	\$ 7,874,800

Notes to financial statements

1. Organization and purpose

The Computer Society of the IEEE is organized within the Institute of Electrical and Electronics Engineers, Inc., an organization exempt from income tax pursuant to Internal Revenue Code Section 501(c)(6). Within the bylaws of IEEE, delegation of the responsibility for the society's operation has been placed with the society's Board of Governors and Executive Committee. The society's constitution states that "The Society shall be scientific, literary, and edu-

cational in character. The Society shall strive to advance the theory, practice, and application of computer and information processing science and technology, and shall maintain a high professional standing among its members. The Society shall promote cooperation and exchange of technical information among its members and to this end shall hold meetings for the presentation and discussion of technical papers, shall publish technical journals, and shall, through its organization and other appropriate means, provide for the needs of its members."

2. Summary of significant accounting policies

Reporting entity

The accompanying financial statements include all society accounts maintained at the society's offices in Washington, DC, Los Alamitos, California, Brussels, Belgium, and certain accounts maintained at IEEE Headquarters. The accompanying financial statements do not include the accounts of society chapters which operate directly under IEEE.

Income recognition

Income from annual membership fees and periodical subscriptions is recognized during the year to which it pertains. The society's share of revenue and expenses for conferences partially or entirely sponsored by the society is generally recognized in the year in which the conference is held.

Fixed assets and depreciation

Fixed assets are reported at cost when purchased. The society provides for depreciation of fixed assets by charges to revenue at rates considered adequate to amortize the cost of such assets over their estimated useful lives (5 to 10 years for office furniture and equipment; 35 years for buildings) on a straight-line basis.

When fixed assets are retired or otherwise disposed of, the property and accumulated depreciation accounts are relieved of the applicable amounts and any profit or loss is reflected in revenue.

Inventory

Inventory consists of tutorial books published by the society and is stated at the

lower of cost or market. Cost is determined on a average cost basis.

3. Investments

Investments consist of unrestricted deposits with IEEE and bear interest based on United States Treasury Bill rates and are carried at cost which approximates market.

4. Fixed assets

Fixed assets as of December 31 are shown in Table 1.

Depreciation expense amounted to \$238,400 and \$226,600 in 1987 and 1986, respectively.

5. Notes payable

Notes payable as of December 31 were as shown in Table 2.

The note payable due May 1, 1990, is collateralized by a first lien on all gross revenues of the society and a mortgage on the Washington, DC, property. Repayment is in graduating amounts through the term of the note with the balance payable on May 1, 1990.

The note payable due May 24, 1988, is collateralized by a deed of trust on one of the two buildings located in Los Alamitos, California. Repayment is in equal monthly payments of \$5,200, including interest, with the balance payable on May 24, 1988.

The note payable due September 25, 1992, is collateralized by certain equipment which was purchased with the proceeds. Repayment is in equal monthly principal installments of \$4,616, plus interest, with the balance payable on September 25, 1992.

Interest expense amounted to \$177,400 and \$165,500 in 1987 and 1986, respectively.

Annual maturities of long-term debt are as follows: 1988 — \$540,200; 1989 — \$86,400; 1990 — \$1,182,700; 1991 — \$55,400; and 1992 — \$41,600.

6. Pension plan

The society is a member of a defined-benefit pension plan sponsored by IEEE. The IEEE plan covers substantially all IEEE employees, including those of the society. It is the policy of IEEE to fund pension costs accrued.

Effective January 1, 1987, IEEE adopted Statement of Financial Accounting Standards (SFAS) No. 87, "Employees Accounting for Pensions." SFAS 87 requires that certain disclosures be made of the actuarial present value of benefit obligations, the projected benefit obligation, the fair value of the plan assets available to meet the projected benefit obligation, and accrued pension costs. Such disclosures are not presented for the society because the structure of the IEEE plan does not readily permit the plan's assets and benefit obligation data to be determined for each individual society. Based upon actuarial valuations of the IEEE plan, assuming a discount rate and an expected long-term rate

The Computer Society of the IEEE Statements of Revenue, Expenses, and Changes in Fund Balance for the years ended December 31, 1987 and 1986

	1987	1986
Revenue:		
Computer Society membership fees	\$ 1,099,300	\$ 720,500
Periodical subscription and other publication activities	8,594,500	7,629,400
Conventions, conferences, and other technical activities	4,574,200	3,951,300
National Computer Conferences distribution	—	140,400
Interest income	118,000	71,400
Other income	322,600	262,000
Total revenue	<u>14,708,600</u>	<u>12,755,000</u>
Expenses:		
Periodical and publication activities	8,418,900	7,467,900
Conventions, conferences, and other technical activities	3,829,500	3,710,200
Administration	1,932,000	1,491,100
Total expenses	<u>14,180,400</u>	<u>12,669,200</u>
Excess of revenue over expenses	528,200	105,800
Fund balance at beginning of year	<u>2,356,300</u>	<u>2,250,500</u>
Fund balance at end of year	<u>\$ 2,884,500</u>	<u>\$ 2,356,300</u>

of return on assets of 8.5% and an increase in the level of compensation of 6.5%, the IEEE plan assets exceeded the projected benefit obligation at December 31, 1987 and 1986.

The society was allocated a net pension credit of approximately \$17,000 in 1987 as a result of the adoption of SFAS 87 and was allocated approximately \$38,000 of pension expense in 1986, based upon its respective share of total payroll and benefit costs.

7. Related-party transaction

Certain general and administrative expenses incurred by IEEE Headquarters and charged to the society amounted to

\$766,400 in 1987 and \$624,000 in 1986. Other transactions undertaken in the normal course of business between the society and IEEE have been reflected in the society's financial statements.

8. Designated fund balance

The Board of Governors has designated a portion of surplus funds received from recurring conferences for use by technical committees in accordance with the society's policy on meeting surplus accounts. The designated amounts are calculated based upon a formula contained in the policy, but in no case can they exceed \$15,000 per technical committee conference.

Table 1. Fixed assets as of December 31.

	1987	1986
Land	\$1,334,400	\$1,334,400
Building and improvements	1,859,700	1,800,800
Warehouse equipment	29,500	25,100
Office furniture and equipment	1,496,500	1,055,900
	4,720,100	4,216,200
Accumulated depreciation	(1,061,400)	(823,000)
	<u>\$3,658,700</u>	<u>\$3,393,200</u>

Table 2. Notes payable as of December 31.

	Annual interest rate	1987	1986
Note payable, balance due on May 1, 1990	Prime	\$1,186,600	\$1,212,200
Note payable, balance due on May 24, 1988	12.0%	456,600	463,900
Note payable, balance due on September 25, 1992	9.5%	263,100	—
		1,906,300	1,676,100
Less: amount due within one year		540,200	33,000
		<u>\$1,366,100</u>	<u>\$1,643,100</u>

ENTER THE 1988 Gordon Bell Awards

The Gordon Bell Awards recognize achievements in large-scale scientific computing. Awards of \$1,000 each will be given in two of three categories:

- **Performance.** A winning program will run faster than comparable engineering or scientific applications.

- **Price/performance.** A winning program must show that its performance divided by system cost is less than comparable applications.

- **Compiler parallelization.** A winning compiler/application must generate the greatest speedup.

The 1987 award winners achieved speedups of 400 to 600 with a 1,024-processor computer.

Entries must be received by Jan. 15, 1989. For complete rules, write Gordon Bell Awards, IEEE Software, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720 or call (714) 821-8380.

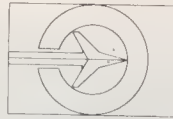
IEEE SOFTWARE

**Late Magazines?
No Magazines?
Membership
Status Problems?
No Answers
To Your
Complaints?**

**Let your
Computer
Society
Ombudsman
cut
through
the red
tape
for you.**



**THE COMPUTER SOCIETY
Ombudsman
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
COMPMAIL +: CS.HELP**



Advance Program*

Fourth Aerospace Computer Security Applications Conference

December 12-16, 1988

Sheraton World Hotel

Orlando, Florida

Sponsored by

The Computer Society of the IEEE

The American Society for Industrial Security

The American Institute of Aeronautics and Astronautics

The Department of Defense Computer Institute

The Aerospace Computer Security Associates

Tutorial Program

Monday 12 December 1988

Secure Systems - An Introduction A Full Day Tutorial

Presented by: Mr. Morrie Gasser, Digital

Data Systems Security A Full Day Tutorial

Presented by: Ms. Teresa Lunt, SRI

Tuesday, 13 December 1988

Recent Developments in Distributed Systems Security A Full Day Tutorial

Presented by: Dr. Harold J. Podell, U.S. G.A.O.

Security Assurance Issues in Configuration Management A Half Day Tutorial

Presented by: Dr. R. Leonard Brown, Aerospace Corp.

Compliance with the Computer Security Act of 1987 A Half Day Tutorial

Presented by: Mr. Ian C. Schreiber, MSM Information Consultants

Technical Program

Wednesday, 14 December 1988

Opening Address: Lt. General B.C. Hosmer, President, National Defense University

Keynote Address: Mr. John J. Lane, Vice President, Computer Sciences Corporation

Technical Panels and Sessions

Track A

Computer Viruses

Luncheon Speaker: Mr. James B. Odom, Associate Administrator, Space Station, NASA

Panel-Computer Viruses

Wayne Weingaertner, NCSC,
Chairman

Panel - DBMS Security

Teresa Lunt, SRI, Chairwoman

Track B

Panel-Trusted Systems Development
Rhonda Henning, Harris InterType
Chairwoman

Trusted Systems Development I

Santosh Chokhani, MITRE,
Chairman

DoD Applications

Track C

Integrity I

Stan Kurzban, IBM,
Chairman

Modeling & Formal Verification

John Millen, MITRE,
Chairman

Integrity II

Robert Jueneman, CSC, Chairman

Reception: 5:15 p.m.

Thursday, 15 December, 1988

Track A

Database Security I
Teresa, Lunt, SRI,
Chairwoman

Database Security II
Kenneth Eggers, MITRE,
Chairman

Database Security III
Patricia Rougeau, TRW,
Chairwoman

Panel-Development Issues for
Future Trusted Systems
Ann Marmor-Squires,
Chairwoman

Track B

Trusted Systems Development II
Terry Arnold, Meridan Group,
Chairman

Trusted Systems Development
Mike Manley, Sun Microsystems,
Chairman

Trusted Network Development
Joyce Nickelson, MITRE,
Chairwoman

Network Security
Dan Schnackenberg, Boeing
Aircraft Corporation,
Chairman

Track C

Trusted Mach as B3 UNIX
Compatible System
Brian Boesch, DARPA,
Chairman

Panel-Computer & Comm Security
in Public & Private Sectors
Ed Sibley, George Mason
University,
Chairman

Intrusion Detection/Protecting
Proprietary Information
Peter Weidemann, SSDS Inc.,
Chairman

NASA Programs
D. Hunt, NASA,
Chairman

Luncheon

**Optional Dinner and Show at King Henry's Feast
Sign-Up on Conference Registration Form**

Friday, 16 December, 1988

Panel-State-of-Practice in
Commercial World
James Wade, Battelle
Chairman

Panel-Policy & PL100-235
Ed Springer, OMB,
Chairman

Industry Panel-Current Issues:
Honeywell, DEC, IBM, Wang
Gerald Isaakson, Information
Security Services, Chairman

Industry Panel-Future Strategies:
Honeywell, DEC, IBM, Wang
Gerald Issakson, Information
Security Services, Chairman

Panel-TNI
Al Arsenault, NCSC,
Chairman

Risk Analysis & Contingency
Planning
Thomas Osgood, CSC,
Chairman

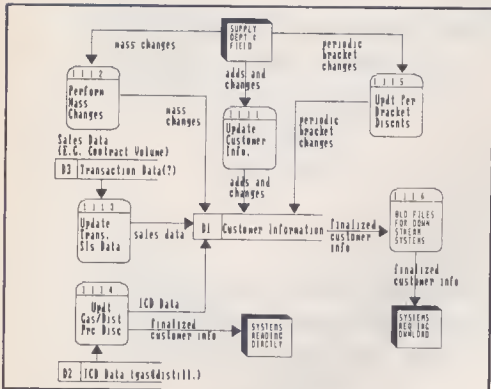
**Optional Disney World Computer Center Tour
Sign-Up on Conference Registration Form**

In Compliance with DOD 5200.12, Dated 16 May 1988, There Will be No Classified Sessions at the Conference.

**For Additional Information, Please Contact
Mr. Robert D. Kovach, Publicity Chairman, at (703) 648-0686,
or Ms. Glenda McBride at COMPMAIL ID: conf.hq**

*** Subject to Change**

WHAT'S REALLY IMPRESSIVE ABOUT OUR CASE TOOLS



GAME AND SARSON DIAGRAM

Beside the price is:

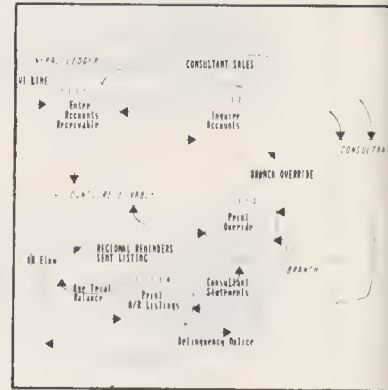
**NO TRAINING REQUIRED —
5 MINUTE INSTALLATION**

Over 4000 Installations Worldwide without one day training or consulting services-- now that's "easy to use"

NO EXPENSIVE HARDWARE REQUIRED

Our modular products are designed to provide the maximum productivity for your MIS budget.

Communications in a standard format
is now possible at a realistic price.



YOURDON DIAGRAM

EXCELLENT UPDATE PROGRAM, SERVICE AND SUPPORT

For only \$50 per tool/per year, you'll automatically receive all updates--that will keep you in tune with the evolution of CASE.

Ever since Computer Aided Software Engineering products came on the market, interest in them has been high. And so, not surprisingly, has the cost of owning them.

But now Visible Systems is offering state-of-the-art CASE technology at an incredibly affordable price.

The Visible Analyst Workbench is the only modular PC based CASE product that runs on the IBM 3270, XT, AT, PS/2 and compatibles. It consists of three integrated tools which can be purchased all at once as a package, or added one at a time as the needs of your organization expand.

It's the quality, flexibility and realistic price of the **Visible Analyst Workbench** that has attracted so many satisfied customers: GE, Shearson, TRW, Met Life, Computer Sciences Corp., The U.S. Government, MONY, NCR, 3M, BellSouth, Touche Ross, Canadair and many more.

THE ABCs OF OUR WORKBENCH.

TOOL A, THE VISIBLE ANALYST

A diagramming tool that allows you to put virtually any organizational idea down on paper quickly, easily and accurately.

With mouse-driven graphics and pop-up menus, it is designed to produce everything from sophisticated systems analysis diagrams

for developing and documenting software systems to the most basic organizational charts used in just about any field.

The Visible Analyst features a custom symbol generator, multiple line styles and text fonts, flexible text processing and editing, line rubberbanding, block move, unlimited nested decompositions and project treefile manipulation.

If that impresses you, just wait until you see our graphics go from your monitor to your printer - with an output of 180dpi, you're going to be looking at real presentation quality.

TOOL B, THE VISIBLE RULES

A package which brings the benefits of structured analysis techniques to the diagramming power of Tool A. It supports the structured methodologies of Yourdon/DeMarco and Gane/Sarson and balances and validates dataflow diagrams over any number of levels.

With The Visible Rules, communication between project members is greatly enhanced, productivity improved and costly errors uncovered much earlier in the lifecycle of a project.

TOOL C, THE VISIBLE DICTIONARY

A fast, comprehensive, easy-to-use tool for database management and system documentation designed for use with the

combination of Tools A and B. The Dictionary database serves as a central definition catalogue that generates detailed documentation, provides uniformity of expression among all analysts on a team, and simplifies future analysis and revision.

The database is automatically populated as each item is drawn. Selective and wildcard searches can be performed. ASCII file formats provided for import/export.

And like the other two tools, its operation is seamless, allowing you to go back and forth between diagrams and dictionary without even taking the time to reload programs.

LAN WORKBENCH

The Visible Analyst Workbench operates on Portables, Standalones and shortly on Novell LANS. Files can be shared across departments or locked as needed. The executable modules can be downloaded to individual workstations as desired to reduce traffic on the net and increase operating speed.

The best news we've saved for last -- only \$595 per tool -- and in quantities of 50 only \$327.25 per tool. Find out why so many companies have decided to grow in CASE with us.

The solution is plainly **VISIBLE**. It's just common sense.

CALL 617-969-4100 or Telex 261102

VISIBLE SYSTEMS CORPORATION

49 Lexington St., Newton, MA 02165

^c 1988 Visible Systems Corporation. Visible, The Visible Analyst and Visible Systems Corporation are registered Trademarks of Visible Systems Corporation.

Reader Service Number 3



The Titan Graphics Supercomputer Architecture

Tom Diede, Carl F. Hagenmaier, Glen S. Miranker,
Jonathan J. Rubinstein, and William S. Worley, Jr.
Ardent Computer

Advanced problems in technical computing, such as fluid dynamics, structural analysis, geophysical interpretation, molecular design, and astrophysics, require extremely high amounts of integer and floating-point computation to perform the analysis required. In addition, high-quality graphics—frequently using dynamics—provide the only effective means to model the initial problem and visualize the results of the computation.

These requirements are typically met by interconnecting a special-purpose graphics terminal or workstation to a mainframe, supercomputer, or minisupercomputer via a local area network or other interface. Numerically intensive computations are executed on the big machine, results are transferred to the graphics workstation, and scenes depicting the results are displayed.

Increasingly, however, a growing number of scientists and engineers require predictable and rapid interactive response for sequences of computation and visualization. While network-linked system configurations have been used to meet such needs, the operational aspects of such configurations can impede—rather than foster—visualization as an integral part of the problem-solving process. These operational factors include high cost, physical

Leading-edge hardware and software technologies now make possible a new class of system—the graphics supercomputer. Titan architecture provides a substantial fraction of supercomputer performance plus integrated high-quality graphics.

proximity, narrow bandwidth between the computing and visualization systems, slow response from time-shared facilities, unpredictable turnaround from batch-oriented resources, constrained functionality from custom graphics hardware, and administrative difficulties.

If an application requires many hours of central-processing-unit time on the largest available supercomputer, using network-linked configurations may be the only choice. However, for applications whose computing needs can be met by a substantial fraction of a supercomputer (or for larger applications where multiple interactive trials are performed before deciding how to expend significant supercomputer time), an interactive, single-user system, combining both high-performance computing and interactive visualization, can be more convenient, more productive, and less costly.

The Titan Graphics Supercomputer is representative of a new class of machine that effectively combines the performance of minisupercomputers with high-speed, high-quality graphics (see Table 1).

This class requires a system with the following characteristics:

- A single, integrated platform,
- A substantial fraction of supercomputer performance,
- Integral high-performance graphics subsystem,
- Integral high-performance hard disk(s),
- Standard network connections,
- Integral visualization,

Table 1. A machine merging a minisupercomputer with graphics.

	Super computers	Minisupers	Graphic supers	Graphics workstations	Workstations
Examples	Cray, ETA	Alliant, Convex	Titan	SGI	Sun, Apollo
Processors	1-8	1-8	1-4	1	1
Sustained Mflops	20-60	1-10	6-24	0.5-1.5	0.5-1.5
Peak MIPS	~ 40	24-36	16-64	10-20	10-20
Interleaving	16-256	4-32	8-16	None	None
Vector features		"Hardware scatter/gather, mask, and compress"		None	None
Vector registers of length 32	16	64	1024	None	None
Peak polygons/sec	None	None	200K	50-120K	1-50K
Image planes	None	None	28-52	16-72	4-32
Selling price	\$1-20M	\$0.1-1M	\$79-150K	\$39-150K	\$9-100K

- Fast interactive response time, in both the computing and modeling environments, and
- Low cost.

This article examines the hardware architecture and implementation developed to meet these design criteria. It describes the key architectural elements in Titan, presents the reasoning behind their selection, explores Titan's processor subsystem, discusses the design constraints and implementation of the I/O subsystem, explains how the processor subsystem reduces the cost and complexity of graphics-specific hardware, describes the Titan operating system, compilers, and graphics software, and concludes with a statistical summary of Titan's performance.

Approach

Preliminary study of key applications indicated the Linpack benchmark¹ showed a high correlation with their performance. A goal of six million floating-point operations per second for the compiled 100 × 100 double-precision version of this benchmark was established for the Titan single-processor entry system. In addition, it was decided that the general-

purpose processors must be fast enough to subsume the special-purpose functional units frequently found in low-cost high-performance graphics terminals (such as display list processors).

Not only were there significant cost savings, but the overall system was simpler. Furthermore, the processing resources of the machine, being non-specialized, could apply to both computation and graphics depending on the load presented by the particular mix of applications being executed. The overall structure of the Titan is shown in Figure 1.

We examined existing machines and a number of system designs capable of meeting the performance goals. High bandwidth connections to memory emerged as a common factor in each of these designs.

Once this bandwidth had been set at a level sufficient to meet the six-Mflops Linpack goal, the maximum system configuration was set. For an application performing concurrent computation and visualization, a configuration with four processors was judged to be the largest reasonably in balance with the available memory bandwidth.

Vector processing was another common attribute of the possible designs. The Don-garra Linpack report of May 2, 1986, contained 16 machines capable of six or more

Mflops on the compiled 100 × 100 benchmark. These machines were all mainframes or supercomputers. The 14 fastest machines (12-46 Mflops) each incorporated a vector processor. The two slowest machines (6.3 and 6.8 Mflops) were mainframes without vector facilities.

Our findings confirmed that the added dimensions of parallelism inherent in a vector processor could permit a single processor—in an affordable single-user workstation—to reach the six-Mflops goal. We, therefore, adopted the fastest available reduced instruction-set computing microprocessor, the MIPS R-2000,² and added multiprocessing extensions and a vector processor of our own design for the Titan CPU.

Early in the development process, it became clear that the fundamental design questions involved were placement of high-speed memory resources to feed the processing elements and assurance of coherence among these resources. RISC microprocessors and vector units have voracious appetites for data. Best performance from each microprocessor implied equipping it with its own instruction and data caches. Sharing caches between microprocessors made sense only if each processor had been so slow or had been so bottlenecked elsewhere that it needed only

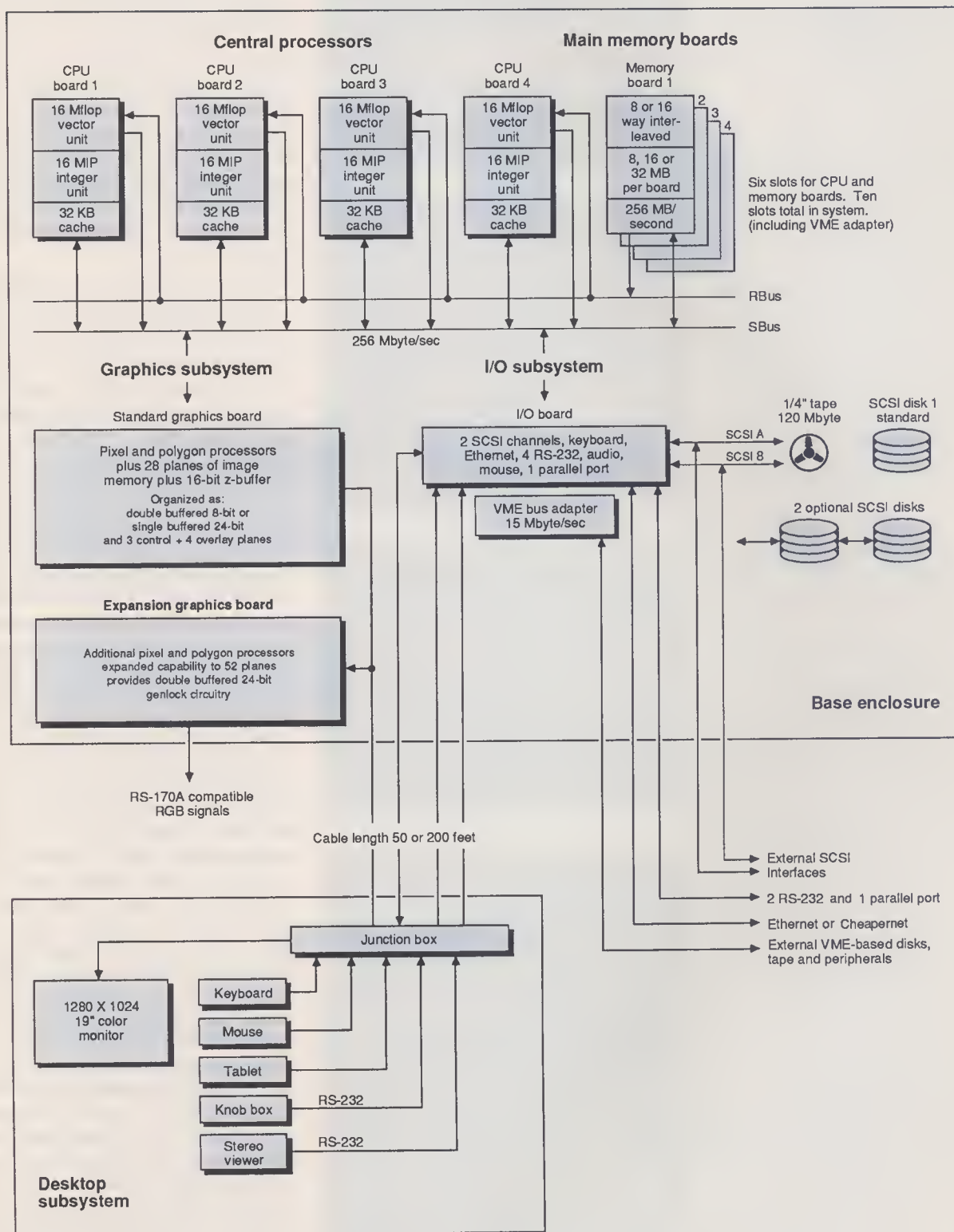


Figure 1. Titan architectural overview.

Titan capable of producing wide-ranging high-quality graphics

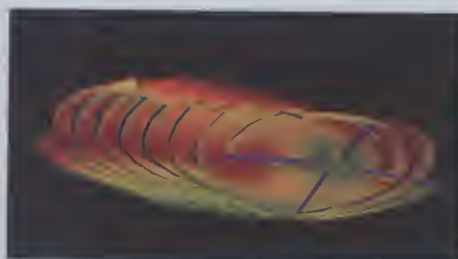
To illustrate the Titan Graphics Supercomputer's high-quality capabilities, four renderings are depicted directly from the Ardent Computer monitor. Each required a different technology for production.

One rendering (a) simulates a helicopter rotor blade slicing through the air, leaving a highly turbulent wake. The changing colors represent differing values of the circulation parameter. By combining computational visualization capabilities in one interactive environment, Titan promotes understanding of complex problems in fluid dynamics. Data is from United Technologies Research Center.

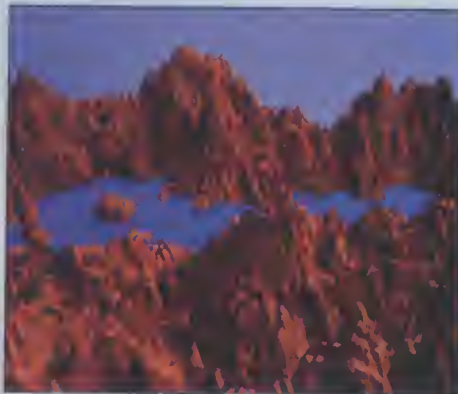
Two DORE (Dynamic Object Rendering Environment) primitive objects are shown in (b): a blue triangle for the water table and a recursively defined triangle mesh for the mountains. A simple elevation threshold function was used to color the vertices either green, brown, or white. The Titan graphics hardware then performed the color interpolation.

In (c), potential flow about a space shuttle orbiter was computed on the Titan by solving a Fredholm integral equation. The solution algorithm computes an LU (lower-diagonal, upper diagonal) factorization of the discretized system, obtained by approximating the equation with linear polynomials. Then, the solution is visualized by mapping surface pressure to hue and dynamically displaying animated particle trajectories along the velocity field.

Data for (d) was obtained from BioDesign and rendered using a molecular viewing program. Chemists needed a variety of rendering styles to focus on different molecular properties. The molecular simulator's visualization environment easily generates different representations from the same molecular data—without recoding. This ball-and-stick image—a 40-atom morphine molecule—could be as easily represented as a wireframe or a ray-traced image.



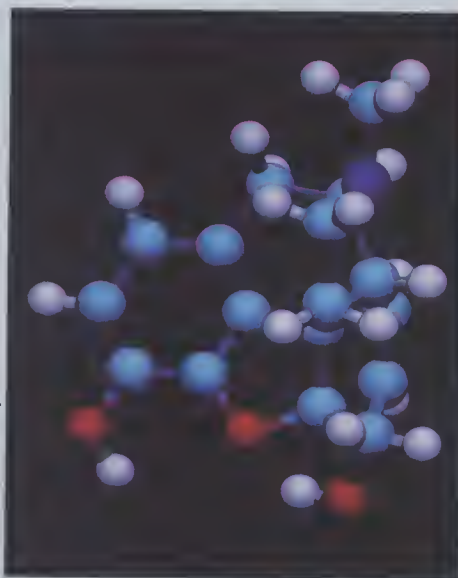
(a)



(b)



(c)



(d)

a fraction of the shared cache bandwidth.

This is not the case with the R-2000 processor. It uses the full bandwidth of its separate instruction and data caches, and any sharing of caches would have caused a performance degradation. The vector units, in turn, each required sufficient bandwidth to sustain concurrent transfer of two streams of data from memory, one stream of data to memory, two streams of data to the computational data path, and one stream of data from the computational data path. This implied the need for a separate vector register file for each vector unit.

In all configurations, the overall design provided a structure in which each computing element has sufficient memory bandwidth to operate at full speed. Preserving this property required an interleaved memory and direct transfer of data between the memory and each cache and vector register file.

After extensive simulation, we rejected alternative designs that loaded or stored vectors through the data caches. Access patterns for the large arrays used in many of the target applications, particularly when non-unit strides are required, can quickly sweep even the largest data caches.

Consider, for example, a vectored inner loop of a Fortran program that traverses the second dimension of one of two $1,000 \times 1,000$ double-precision floating-point matrix operands and stores the computed results in the second dimension of a third $1,000 \times 1,000$ matrix. A single pass through the loop would (if there are no cache conflicts among the data) overwrite more than 3,000 cache lines. For caches with line sizes of 8, 16, 32, 64, or 128 bytes, respectively, this single pass would read 16,000, 24,000, 40,000, 72,000, or 134,000 bytes of data from memory. Writing the result would cause an additional respective 8,000, 16,000, 32,000, 64,000, or 128,000 bytes of cache to be invalidated for a write-through cache, or to be filled from memory for a write-back cache.

For cache lines that have multiple cycle fill times, it is also slower to process vectors with non-unit strides since it is necessary to wait for unneeded data to be loaded. Figure 2 shows the effective memory bandwidth for a large stride (> 16) access of an array of double-precision numbers and different cache line sizes.

Even if this extra data is subsequently referenced—assuming it survives sufficiently long in the cache—the overall processing is slower than if the data had been supplied directly to the vector unit from

the memory at the rate of one word per cycle.

Coherence among caches and vector register files was achieved by adopting a cache-buffered write-through protocol and bus-watchers; the details of this will be described later in this article. A uniform virtual-addressing model, covering the integer processor and its associated vector unit, was realized by incorporating an external translation look-aside buffer (TLB) for data transferred between memory and the vector register files. This will also be described later.

Integer processing unit

Titan was designed to provide the highest vector performance possible in a price-constrained, single-user system. To meet this goal, Titan's integer unit had to be capable of issuing instructions to the vector unit at a high enough rate to ensure that it would never be idle. It also needed enough speed to support integer calculations at a rate that would provide a well-balanced machine. These requirements, together with the modest clock rates possible with the CMOS implementation technology we chose, implied a processor that had a low number of clocks per instruction. A RISC processor, with an interface capable of controlling a tightly coupled co-execution unit, met these requirements.

The integer processor. The core of the integer processing unit (IPU) is shown in Figure 3. By selecting the fastest integer processor currently available, we were able to focus on the floating-point, memory, I/O, and graphics portions of the machine. Using commercially available components allowed us to concentrate design efforts in those areas where we could make unique contributions, and also helped us reduce costs.

The R2000 is a 32-bit RISC processor with a load/store pipelined architecture capable of executing one instruction every clock cycle. A single chip contains the processor plus a 64-entry, content-addressable TLB for memory management and full cache control. Logically, the bus interface to the chip is a split-bus architecture with separate paths for instruction and data accesses. Physically, these buses are time multiplexed at twice the operating frequency to reduce pin count. With this processor operating at 16 megahertz (as it does in the Titan) and using the Titan cache architecture, the average instruction

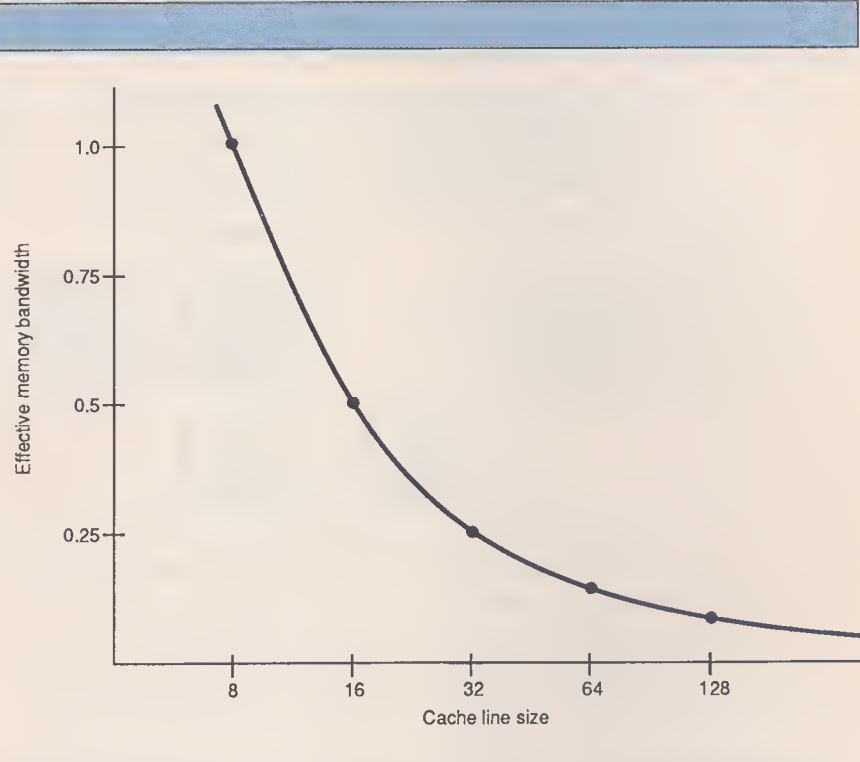


Figure 2. Effective memory bandwidth versus cache line size.

execution rate is 10 million instructions per second.³

Integer processor cache. Titan uses 32 kilobytes of cache for each R2000 processor. This cache is split into 16 kilobytes of instruction cache and 16 kilobytes of data cache. Since each cache can cycle in one clock, the processor can access both caches during one clock cycle, providing an aggregate transfer rate of 128 megabytes per second.

While the standard R2000-based system uses a line size of one word, Titan achieves improved performance by using a line size of two words to increase the cache hit rate. Since the time to fetch two words from memory is 16 cycles (compared to 15 for a single word), Titan reduces the average time to do a cache fill from 15 to 8 cycles by doubling fetches.

Potentially, the write-through write strategy for the data cache can slow down the processor because all writes are misses. However, using a four-deep write buffer ensures that the processor rarely stalls on write accesses. This buffer allows the processor to execute writes in one cycle, stalling the processor only when the buffer

is full and the processor executes an additional write.

The size of the buffer reflects an attempt to minimize gate count while allowing enough depth to minimize the performance impact of the write-through strategy. (Over a broad range of applications, write stalls add about 0.1 cycles per instruction.) The buffer is organized as a fall-through FIFO queue.

When a data fetch misses in the cache or is an uncached data access, the read is stalled until the write buffer is emptied. Naturally, instruction fetches bypass the write buffer and are allowed to proceed ahead of the queued writes. Therefore, except for stores into the instruction stream, the queuing of writes is transparent to the software. For programs that rely on self-modifying code, instructions are provided that explicitly wait for the write buffer to empty and that purge the instruction cache.

Multiprocessing support. Because Titan is a symmetric multiprocessor that allows task decomposition and migration between processors, maintaining cache coherence between processors is critical.

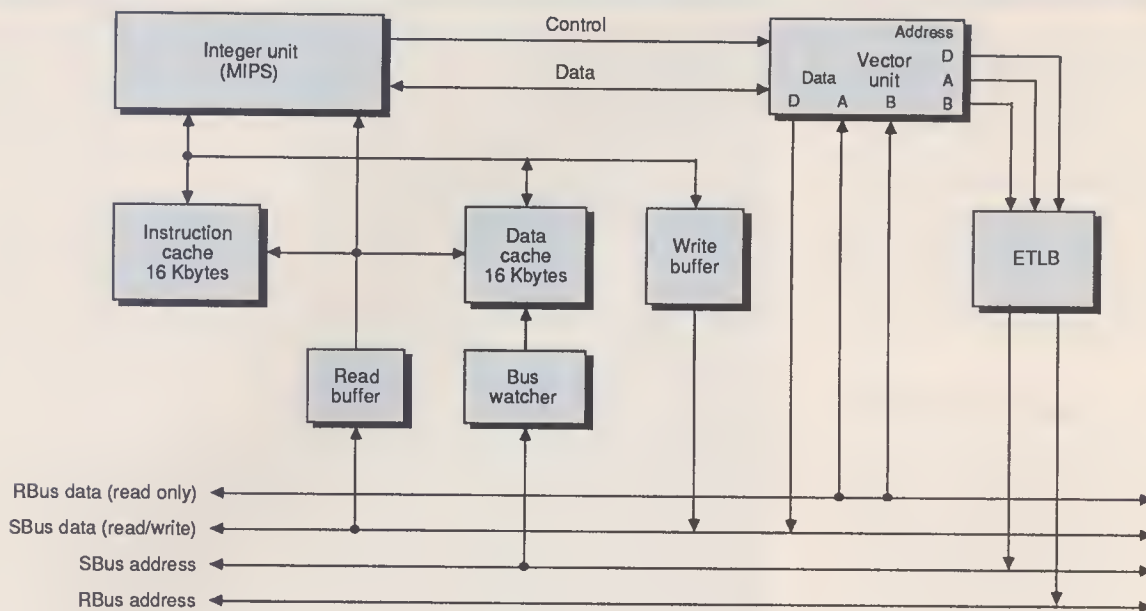


Figure 3. CPU block diagram.

Cache must also be kept consistent with memory because of concurrent operation of the vector units, I/O, and graphics subsystems.

Data cache coherence in Titan is maintained by a bus-watching scheme.⁴ This scheme is attractive because it requires no bus overhead or retry mechanisms (which are easy for a scalar processor to perform but unacceptably inefficient and complex in a vector-processing environment).

Typically, the drawback of a bus-watching scheme is that the directory of entries in the cache—or cache tags—must be shared between the processors and the bus. This causes performance degradation when both attempt to access the cache simultaneously. In Titan, we ensured minimal degradation by using a duplicate set of tags to prevent this conflict.

Each processor contains its own bus watcher that monitors the Titan bus for all transactions. A write on the bus whose address matches an entry in the bus watcher tag set causes the entry in the bus watcher and in the data cache to be invalidated. Thus, the only accesses made to the processor's data cache from the bus watcher occur when there is a hit in the bus watcher tags. This significantly minimizes the frequency of interference. Instead of

being updated, the data cache is invalidated in an attempt to minimize cost. In addition, updating the cache would require an additional data path from the bus to the cache.

One notable aspect of Titan's cache coherence mechanism is that the bus is always allowed to run at full speed and is never required to stop or retry. Should a hit occur in the bus watcher, the data cache invalidation overrides the processor if it happens to be executing a data write during the invalidation cycle. If a data cache read occurs, an invalid entry is returned forcing a retry; otherwise, the processor is not affected.

This capability is an important feature. It maximizes performance and greatly simplifies the design of the other writers of memory in the system, since rerunning deep pipelines (especially in the vector unit) is a complicated and expensive process and is never needed by the cache-invalidation protocol.

In addition to a cache coherence scheme, multiprocessor systems also need some form of hardware synchronization primitives. Moreover, to support fine-grain parallelism, a low-overhead, user-accessible synchronization mechanism is essential. Titan supports two types of

hardware synchronization: semaphores and doorbells.

Two types of semaphores are available. A test-and-clear instruction supports simple semaphores, and a test-and-increment instruction supports counting semaphores. A pair of registers supports these hardware semaphores. One stores the virtual address of the semaphore. The other, when read, generates a special semaphore bus cycle and returns the data from the memory address contained in the semaphore address register.

Note that the address stored in the semaphore address register is a virtual address, requiring translation before it is emitted on the bus. Titan provides this address translation by using a 4K-entry, direct-mapped TLB. In addition, semaphore accesses are treated in the same fashion as external writes by the bus watchers, ensuring that the cache and memory are consistent. This is an important feature because semaphore accesses bypass the cache.

Doorbells, the other synchronization mechanism, allow any processor to interrupt any other processor. This mechanism is used by the operating system for inter-processor communication and is implemented with a broadcast protocol on the system bus.

The Titan bus

A system with vector processing capabilities requires very high bandwidth to memory so that the data transfer rate does not limit its computational speed. In a system like Titan, where processors communicate with memory over a bus, it is critical that the bus not become a bottleneck. The Titan bus was designed with this in mind.

Simulations of the bus protocols, under the heavy loads expected from vector applications, were performed to insure that the bus would operate at high levels of utilization. In addition, the IPU caches were placed between the processors and the system bus. As a result, only IPU memory references that miss in the cache generate bus traffic.

The Titan bus has four logical pieces: two data transfer buses, Read and System; an arbitration bus; and a control bus carrying clocks, interrupt lines, reset lines, and the like. The bus signals are transistor-transistor logic-level except for the clock distribution and arbitration control signals that are emitter-coupled logic. The Read, or R, bus is so named because it is used exclusively for vector read traffic. The System, or S, bus is used for all other transactions, including vector writes and both read and write accesses by the integer processors, I/O, and graphics. Figure 1 illustrates the two bus complexes and their interconnection with the various components of the Titan system.

Each data transfer bus consists of a 32-bit address bus and a 64-bit data bus. Each bus runs at 16 MHz and obeys a synchronous, split-transaction protocol. With 16 bytes of data transferred every 62.5 nanoseconds, the sustained transfer rate of the Titan bus is 256 megabytes per second. For a given transaction, the address bus and the data bus may be used at different times.

For writes, a fixed delay exists between the time the address for a transaction appears on an address bus and the time the data for the transaction appears on the data bus. However, the response time for reads is variable, even though bounded by a time-out mechanism. Requests are tagged with a unique requester ID so that the returned data can be correctly routed.

Both buses are used for memory reads. Only one bus, the S bus, is used for writes. There are two reasons for this. First, simulation showed that allowing writes on both buses did not significantly increase system performance. Second, this restriction means the bus watchers must watch only

The memory subsystem can have multiple requests pending simultaneously from one or more requesters.

one bus. Because graphics and I/O writes to memory also take place on the S bus, cache coherence for these transfers is automatically maintained by the same hardware used for processor writes.

All devices that use a bus must arbitrate for a free slot on that bus. For a read or write request, an initiator requests a slot on an address bus in one cycle and, if there are no conflicts, receives the slot in the next cycle. Arbitration is single cycle and is re-arbitrated every clock cycle. By design, there is no contention on the bus, even with back-to-back transfers. No cycles are wasted for bus hand-off.

A request may not be placed on the bus unless the target resource is free. Resource status is denoted by a backplane signal for each memory interleave, graphic, and I/O. This restriction ensures that no bus cycles are wasted on a request that cannot be honored by the responder without requiring a busy responder to queue requests.

The priorities on the bus are a combination of fixed and round robin. Each class of requester has a fixed priority. Where multiple requesters exist in a class, a round-robin priority scheme is used. In machines with interleaved memory and multiple vector memory pipes, inefficient access patterns, known as linked conflicts, can arise. These patterns are repeating sequences of conflicts between multiple vector streams that result in nonoptimal steady-state performance.⁵ The Titan arbitration scheme is designed to minimize these conflicts.

To ensure high reliability, all noncontrol bus signals have byte parity. In addition, handshakes are designed to ensure acknowledgment of all transactions. Errors are always signalled to the initiator of the failed transaction, when the identity can be determined unambiguously. Otherwise, the processor with the lowest serial number is signaled.

Memory subsystem

The Titan memory subsystem is designed to provide sufficient bandwidth to support the integer units, vector units, I/O subsystem, and graphics subsystem. It is capable of delivering data to both the R and S buses every clock, allowing the system to sustain a memory throughput of 256 megabytes per second. Each memory board can contain 8, 16, or 32 megabytes using one-megabit dynamic RAMs; and 32, 64, or 128 megabytes using 4-megabit DRAMs. A maximum of four boards may be used in the system, providing a range of 8 to 512 megabytes.

Each memory board is eight-way interleaved. Each of the eight interleaves can be accessed independently. This allows all eight interleaves to service requests simultaneously. The memory boards operate in pairs to provide 16-way interleaving whenever two of the same-sized boards are installed. Memory in each interleave is physically organized 32 bits wide, and interleaves are organized such that any access from the bus is serviced by one interleave. Double word accesses are supported by using static column mode DRAMs to provide 64 bits of data in two clock cycles. This organization requires each interleave to be addressed on double-word boundaries. In a sustained sequence of read operations, the data returned on the bus each cycle would consist of bits 31-0 (high-order word) from the i th, together with bits 63-32 (low-order word) from the $(i + 1)$ st read.

Memory is protected by an error-correcting code with single-error detection and correction, and double-error detection. Because the DRAMs used are organized four bits wide, the ECC scheme also detects any single DRAM chip failure from one to four bits. The error is logged, and the corrected data is returned on the bus. However, the corrected result is not rewritten to memory; instead, the operating system periodically issues a "scrub" transaction to each memory address, at which time the corrected data is rewritten. Nonetheless, the corrected data is immediately rewritten for any transaction requiring a read-modify-write cycle.

The memory subsystem can have multiple requests pending simultaneously from one or more requesters. It guarantees that data from the read requests will be returned in the same order the requests were received. This property greatly simplifies the design of the address generators of the vector unit as well as the bus pro-

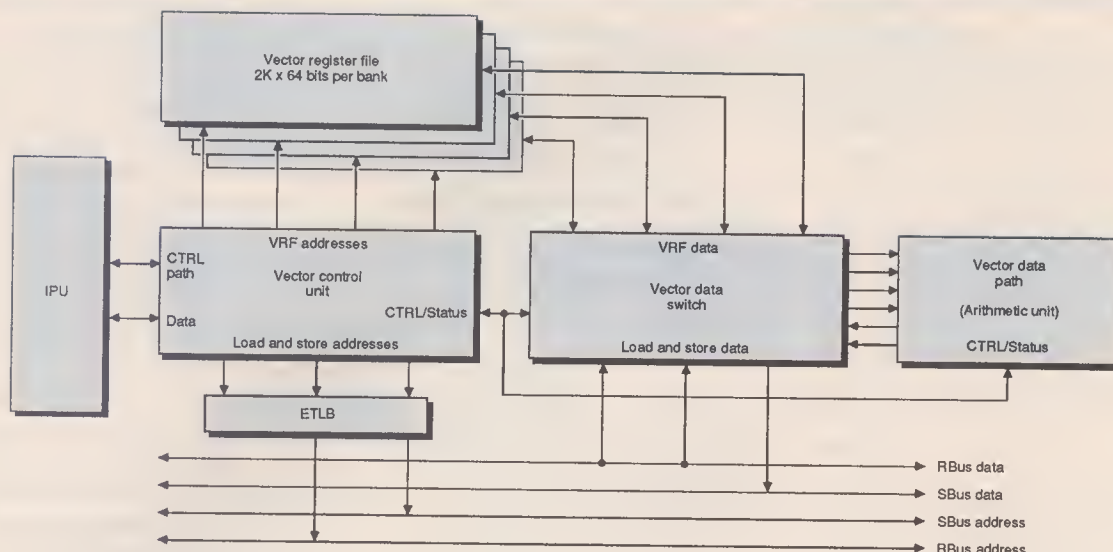


Figure 4. Vector processing unit block diagram.

tol because read responses need only be tagged with the ID of the initiator of the read request and no reordering or queuing of returned data is required.

Titan vector unit

The Titan vector unit executes all floating-point instructions, both vector and scalar, as well as vector integer arithmetic. While an independent scalar floating-point unit would have been useful, cost and space limitations were prohibitive. Thus, just as in the Cray X-MP, the same floating-point functional units are used for both scalar and vector operations.

The pipeline management hardware needed to execute vector instructions is also applied to the execution of scalar instructions, however. This allows multiple scalar instructions to be executed concurrently. The scoreboarding technique differs from the one used by the CDC6600⁶ or the 360 Model 91 floating-point unit⁷ and is examined below in some detail.

The design goals for the vector unit were

- *Low start-up times.* Scalar instruction rates and the effectiveness of short vector

instructions increase in inverse proportion to start-up time.

- *Maximal intrapipeline and interpipeline overlap.* Keeping pipes full and keeping multiple pipes running concurrently maximizes the number of operations retired per clock tick and, thus, maximizes the computation rate.

• *Graphics support.* The vector unit has several features expressly included to guarantee fast execution of graphics code.

- *Fast context switching.* Because Titan supports interactive computation, the ability to change quickly among a number of processes is critical.

The vector unit, shown in Figure 4, has five major components organized around a crossbar switch:

- A vector register file (VRF), which stores up to 8K double-precision words.
- Three independent arithmetic functional units: ALU, multiplier, and divider, with an aggregate peak computation rate of 16 Mflops.
- Three independent memory pipes: two load pipes, each capable of running at eight million accesses per second, and one store pipe capable of running at 16 million accesses per second.
- A crossbar switch for routing data between the system bus, the arithmetic

functional units, and the VRF.

- Translation look-aside buffers to translate the virtual memory pipe addresses to physical addresses.

- Interface to the integer processing unit.

Vector register file. With the notable exception of the CDC Star, Cyber 205, and ETA machines, all vector processors* are register-to-register operation machines.¹ While the CDC line demonstrates that memory-to-memory vector architectures can be effective, the register-to-register organization has two important attractions:

- (1) The operand bandwidth required by the functional units of a vector machine is very high and is provided more economically by a small, fast resource like a register file than a large, slow resource like memory. For example, consider a machine with two arithmetic functional units: a multiplier and an adder. To support the peak execution rate of these two units, a six-ported operand source/sink is

* The IBM 3090 vector unit does have some register-to-memory vector operations. However, with the exception of some small code-density advantages, it is equivalent to an operation chained to a load.

required. If the operands are obtained from memory and written to memory and we wish to have integer or I/O operations proceeding concurrently with vector operation, still more memory bandwidth is needed. This, in turn, requires additional memory ports, additional memory interleaving, or a physically more highly partitioned memory.

(2) Though at the macro level reuse of operands in numerically intensive computations is quite low, increasingly sophisticated compiler technology will automatically restructure code to park array segments in registers. This further reduces memory bandwidth requirements.⁸⁻¹⁰

Titan's vector unit has four independently operating pipelines. To provide sufficient bandwidth for each pipe to run at peak rate, we needed a VRF capable of supporting 56 million double-precision accesses per second, or 448 megabytes per second:

- Eight million accesses per second on each of the two memory load pipes.
- 24 million accesses per second on the operation pipe, based on the fact that the maximum valence of any vector operation is three.
- 16 million accesses per second on the memory store pipe.

Furthermore, for linked vector dyads** to run maximally overlapped, the VRF must be logically at least six-ported: one port for each of the two loads, two ports for each of the two input operands, one port for the operation result, and one port for the store.

In Titan, we did not take the conventional approach of using a multiported register file as the storage elements in the VRF. Instead, we used a small, high-speed static RAM organized as four independent banks. Each bank is organized eight bytes wide and 2,048 bytes long, with a bandwidth of 16 million accesses per second. Each of the 2,048 locations is known as a cell and holds an operand, either single precision or double precision.

The VRF is addressed using a two-part, 13-bit cell address: $VR < 12..0 >$; where $VR < 12..11 >$ specifies the bank and $VR < 10..0 >$ is the offset into the bank.

The logical organization of the VRF can

be given a finer structure by addressing convention. For example, if we agree to specify offsets that are multiples of 32, and agree not to specify vector operations of a length greater than 32, the file appears as 256 vector registers of length 32. Note that it is the addressing convention—rather than physical arrangement—that gives the file its organization. In Titan, this addressing flexibility is reflected in the architecture, rather than by some fixed convention in the hardware.

By convention, the operating system divides the VRF into eight pieces, or frames, each of length 256 cells. This yields up to eight user sets of vector registers, each with 32 registers of length 32. This organization supports Titan's goal of serving as an interactive visualization tool. These independent user VRF sets reduce the frequency with which the VRF has to be saved on context switches, improving responsiveness of the machine in an interactive environment. A simple-base and upper-bound protection scheme keeps users from addressing outside their frame.

An operand specifier in Titan is a 10-bit quantity: a two-bit bank specifier and an eight-bit bank offset. Up to three such specifiers may occur in a Titan macro order, which is 42 bits comprising a 10-bit opcode and a 32-bit operand specifier.

Note that while, in aggregate, the VRF is organized as an eight-ported, one-gigabyte-per-second register file, this bandwidth and effective porting are achieved only if four different banks are accessed on a given cycle. If there is bank conflict, an arbitration scheme determines which cell-access requests are honored. Any request that is preempted causes the pipe making the request to stall. The arbitration is done on a fixed-priority scheme. In decreasing order, the priorities are as follows:

- Both load pipes. (Since the aggregate load bandwidth is 16 million stores to the VRF and each bank can support 16 million accesses per second, this ensures that—in the worst case—returning data from memory can be accepted without the use of buffers.)
- Operation pipe write-back requests.
- Operation pipe fetches.
- Memory store pipe fetches.

We assigned highest priority to the load pipes to avoid having to provide buffering for returning memory data. The other assignments were made to optimize throughput based on simulation of a large number of VRF access patterns.¹¹

Performance of this VRF organization is sensitive to the compiler's ability to make operand VRF bank assignments that minimize bank conflict among concurrently executing operations. This was one of the central vector compilation challenges for Titan, and it has been effectively solved. The large number of vector registers and the regularity of the VRF organization substantially eased the difficulty of the problem. For example, in the set of 24 Livermore Loops, none have any VRF bank conflicts.

There are several additional important benefits from the organization of the VRF as a small, interleaved memory, addressable on the cell boundary. Cell addressability allows computations such as convolutions to be coded directly. An example, shown in Figure 5, is a simple assembly code fragment that computes a third-order convolution. An additional property of the Titan vector unit (discussed in detail in the section entitled "Scoreboard") is that all operations, including the individual elements of a vector operation, are executed in an order consistent with sequential execution. This feature, coupled with individual cell addressability, allows recurrences to be executed directly.

In Figure 6, we show a program fragment for computing the Fibonacci series as a single vector instruction (the code assumes that the first two locations in the vector register, V1, have been set to 1). A final advantage is that the hardware implementation can employ large, fast SRAMs, rather than smaller, more expensive, multiported register files.

Memory interface. To move data between memory and the VRF, Titan has two load pipes and one store pipe. We opted for a set of three memory pipes since this allows even fully memory-linked operations (both inputs and the output) to run at full speed. (Two pipes would not allow peak execution of vector dyads nested in a loop.)

Each load pipe can fetch single-precision or double-precision operands at eight million accesses per second. Read requests are issued asynchronously to the memory. Values are returned from the memory in the order the read requests were issued; no tagging or reordering logic is required. Each load pipe supports constant stride accesses at a rate of eight million accesses per second, as well as gather operations at a maximal rate of 5.3 million gather accesses per second.

** A linked dyad is a vector operation taking two input vectors and producing one output vector, with the source and result operands residing in memory.

```

DO 1 = 1,32
  Y(1) = W(1)*X(1)
        + W(2)*X(i + 1)
        + W(3)*X(i + 2)
ENDO

; X is in v0, Y is stored in v4
; W is in v1

li    $at, 32
sw    $at, <vlength> ; set vector length to 32
dvml  $v2, [$v1.0], $v0.0 ; v2 = W(1)*X(i) {length 32 vector op}
pass  $a0, $v1.1 ; ac0 = W(2) {scalar op}
dvma  $v3, $a0, $v0.1 $v2 ; v3 = ac0*X(i + 1) + v2 {length 32 vector op}
pass  $a0, $v1.2 ; ac0 = W(3) {scalar op}
dvma  $v4, $a0, $v0.2, $v3 ; v4 = ac0*X(i + 2) + v3 {length 32 vector op}

```

Figure 5. Assembly code fragment for third-order convolution.

```

li    $at, 23
sw    $at, <vlength> ; set vector length to 32
dvadd $V1.2, $V1.1, $V1.0 ; v1[2] = v1[1] + v1[0]

```

Figure 6. Assembly code fragment for Fibonacci.

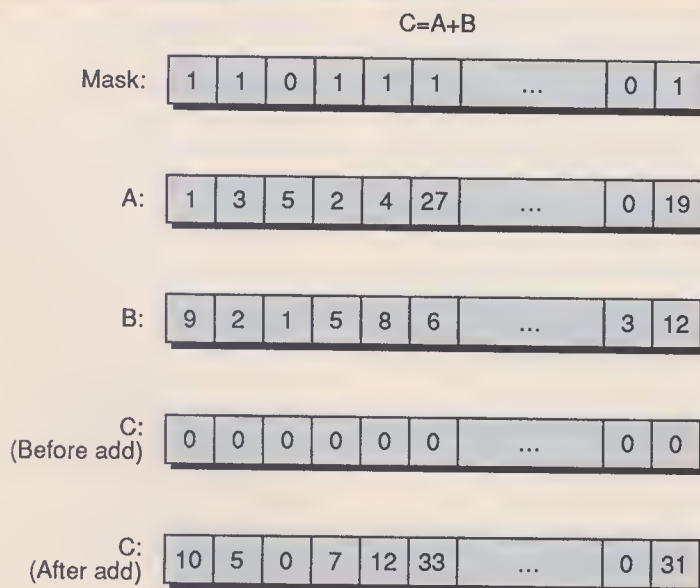


Figure 7. Operations under mask.

The single store pipe has a maximal rate of 16 million stores per second. It supports constant stride stores at a rate of 16 million per second and scatter store operations at a rate of 5.3 million per second.

The memory pipe bandwidth is sufficient to support full pipe-rate (eight megahertz) execution of linked vector dyads. The store pipe's double rate allows it to catch up when suboptimal VRF allocation forces it to stall during the course of a computation. It also reduces memory traffic by storing adjacent single-precision results using a single (double-precision) memory request.

Titan has a virtual-machine architecture. The address sources for the memory pipes produce reference streams in virtual space. To translate these streams to physical addresses for requests to memory, a pair of 8K-entry, direct-mapped external TLBs, each mapping a four-kilobyte page, are provided. The ETLBs are each associated with a specific bus, allowing simultaneous accesses for both load and store pipes. In addition, this association allows address translation concurrently with bus arbitration, so that memory-pipe latency is not increased by translation. The integer processor TLBs and the vector unit ETLBs are maintained by the operating system in a manner that assures mutual consistency.

Arithmetic operation units. Titan has three independent floating-point functional units. There is an eight-Mflops arithmetic logic unit, an eight-Mflops multiplier, and a one-Mflops divider. Computation is done on single-precision and double-precision IEEE-format numbers and 32-bit 2s complement integers. All operations of the vector unit use the vector-register file as the source and sink of operands. In addition to simple addition and multiplication operations, Titan performs a variety of explicit reduction and multiply/accumulate instructions.

Comparison operations have a 256-bit mask vector as the result target. There is one bit in the mask for each element address in the VRF. The *i*th element of a compare operation, depending on the selected condition and the operation result, will set or clear the corresponding mask bit.

To support efficient compilation of conditional programs into vector code, arithmetic and memory operations can be executed conditionally, using the mask. The effect of doing an operation under mask is shown in Figure 7. Every element

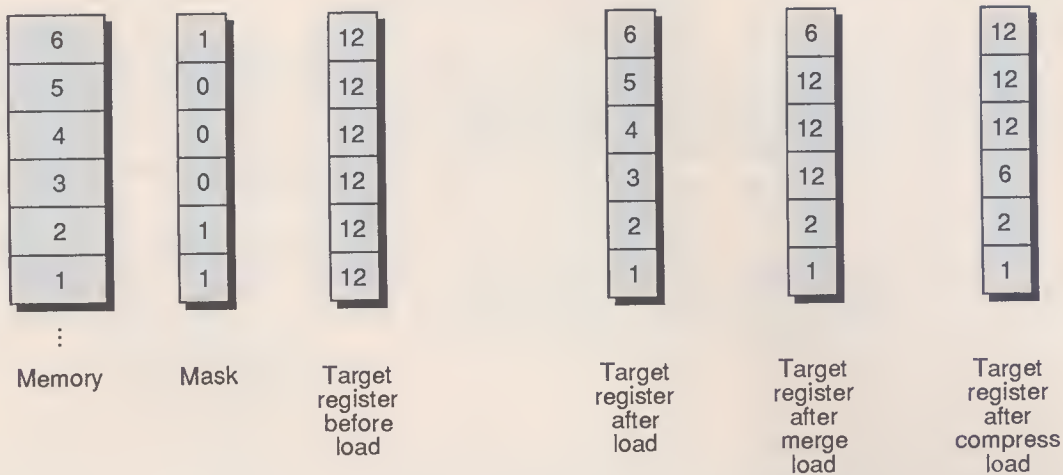


Figure 8. Masked load operations.

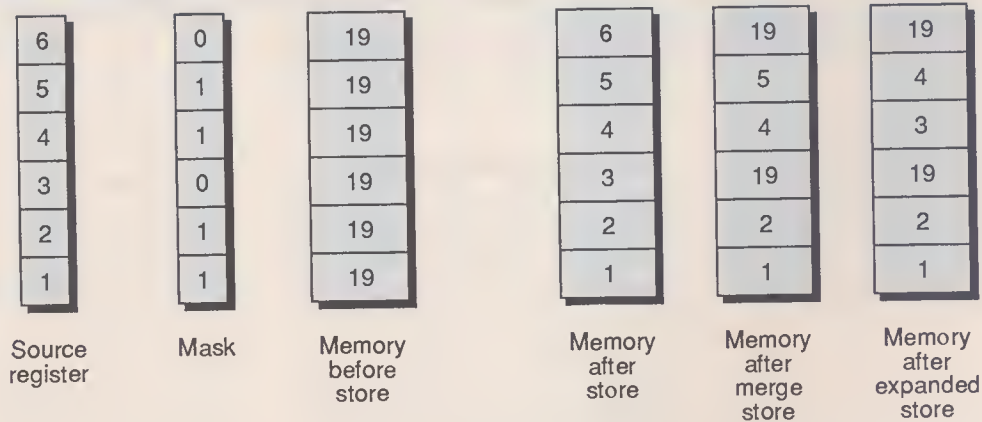


Figure 9. Masked store operations.

of the vector operation that corresponds to a "0" bit in the mask or to a "1" bit if using the complement of the mask, has no side effect. The destination is not modified, and exceptions are not detected.

For memory loads under mask, only the elements corresponding to "1" bits in the mask are fetched from memory. These are either merged or compressed into the destination register, as shown in Figure 8. For memory stores under mask, only the elements corresponding to "1" bits in the

mask are stored. These are either merged or expanded from the source register, as shown in Figure 9.

Scoreboard. The vector unit contains four independent pipelines: the three memory pipes and the operation pipe. Each pipe can have a different instruction (or element of a vector instruction) pending in each of its stages. Thus, between concurrency within each pipe, and between the four pipelines, nearly 60 por-

tions of vector instructions can be in process at a given moment.

Concurrent execution of instructions can yield results that are dependent on the order of execution. Consider concurrent executions of two instructions issued at time t_n and t_m (t_m later than t_n , both referencing a common VRF cell). If the first instruction produces a result, A, needed by the second instruction, the second instruction is said to have a chain dependency upon the first. If this is the

case, execution of the second instruction must be delayed until the first completes its write of A. Completion of this write is called clearing the conflict. If both instructions write the same location, then a data hazard exists and the instructions must complete in the order issued. Finally, if the first instruction reads a location that the second instruction writes, a data hazard exists and the instructions must proceed such that the read of A by t_n precedes the write of A by t_m . These data conflict types are summarized in Table 2.

In general, a single-vector instruction specifies a sequence of operations that accesses many cells. Thus, two vector instructions may go in and out (or out and in) of conflict as their execution proceeds.

Although there is a large amount of concurrency, the Titan programming model is simple. Hardware hazard and chain detection for data conflicts involving the vector register file ensure that all operations are completed in an order consistent with sequential execution. That is, the observable results of the identical computations will be those that would result if each computation proceeded serially, one after the other. The scoreboarding algorithm guarantees such consistency while achieving maximal overlap in the actual processing.

The benefits of this overlap are substantial, although the impact on vector performance is higher than on scalar performance. For example, Titan performance on the 24 Livermore Loops would have been nearly 25 percent lower had no overlap been allowed between the four

pipelines. Linpack performance would have been less than half Titan's present performance.

Scalar floating-point computation. Scalar floating-point operations are executed in Titan's vector unit. They are treated essentially as unit-length vector operations. Rather than having a special set of scalar registers, the first eight locations in each VRF bank are the sources and destinations for all scalar operations and scalar-memory loads and stores. Since they are located in the VRF, the scalar locations may also serve as operands for vector operations.

Several advantages accrue from this approach:

- Substantial hardware savings are enjoyed, since only one set of floating-point arithmetic units is required.
- Mixed vector/scalar operations have short start-up times.
- The hazarding and chaining hardware used to allow concurrent execution of vector operations provides the same capability for scalar operations. As a result, sequences of data dependent scalar operations may be issued without regard to operation latency, since they will be executed in the correct order. Independent scalar operations can complete at peak vector rates.

Vector floating-point performance. The result of a low-instruction overhead, together with the ability to overlap strip housekeeping computations in the integer unit and vector computation in the vector unit, is a high operations rate, even for relatively short vector lengths. Two ways to characterize a vector machine are by the maximum computation rate achieved for a given operation and the length of a vector at which half that computation rate is obtained. These two numbers are referred to as r_∞ and $n_{1/2}$.^{12,13} Performance figures for the Cray X-MP are also shown in Table 3.¹⁴ Performance is reported for rates achieved in a simple Fortran loop containing the indicated operation.

Table 2. Hazard and chaining definition.

Action	Later	
	Read A	Write A
Read A	<i>no conflict</i>	<i>hazard</i>
Write A	<i>chain</i>	<i>hazard</i>

Table 3. Performance figures for the Cray X-MP.

Operation	Titan		Cray	
	r_∞	$n_{1/2}$	r_∞	$n_{1/2}$
$v = v + s$	7.48 Mflops	7.7	76.6 Mflops	~ 15
$v = v * v$	6.32 Mflops	7.3	68.7 Mflops	~ 25
$v = s + v * v$	11.68 Mflops	6.7	102 Mflops	~ 13

I/O subsystem

Titan's I/O subsystem has several goals. To support high computation rates, it must deliver high aggregate data rates, with low latency access. To succeed as a single-user computer, it must be low cost and allow connection to low cost devices. In addition, it must provide a flexible means of providing system expansion.

Meeting the I/O goals. Because the Titan system bus provides such high performance, the I/O bus (shown in Figure 10) was designed as a scaled-down copy of it. This synchronous, split-transaction bus provides an aggregate transfer rate of 64 megabytes per second and can deliver 32 megabytes per second to an individual requester. A large number of direct memory access controllers, totalling 11 channels, are coupled to the I/O bus, providing an architecture that allows each device controller to be designed as if it had a direct path to system memory. A simple address translation scheme further increases the flexibility of the approach by providing scatter/gather capability on each DMA channel. This capability is used by the Titan's Unix fast-file system to transfer data directly to and from user memory. This approach significantly reduces latency and operating-system overhead. This I/O architecture can provide the necessary performance and, yet, minimize the subsystem cost. Because each device is capable of accessing system memory directly, very little buffering (less than 100 bytes) on the I/O boards is needed. Multiple, self-contained channels, scatter/gather address translation and low interrupt latency combine to make a separate I/O processor unnecessary.

The Titan I/O subsystem provides a reasonably full complement of devices and interfaces in the base configuration. To ensure that a wide variety of devices and interfaces could be supported, a standard public bus—implementing the VMEbus standard—is also included.

Choosing the disk interface. Looking at trends in the disk drive industry convinced us that the 5¼-inch drive format is, and will continue to be, the most cost-effective format for our performance and capacity needs. We had two interface choices: enhanced small-disk interface and small computer system interface.

ESDI had two advantages. As a lower-level interface, it readily provided hooks

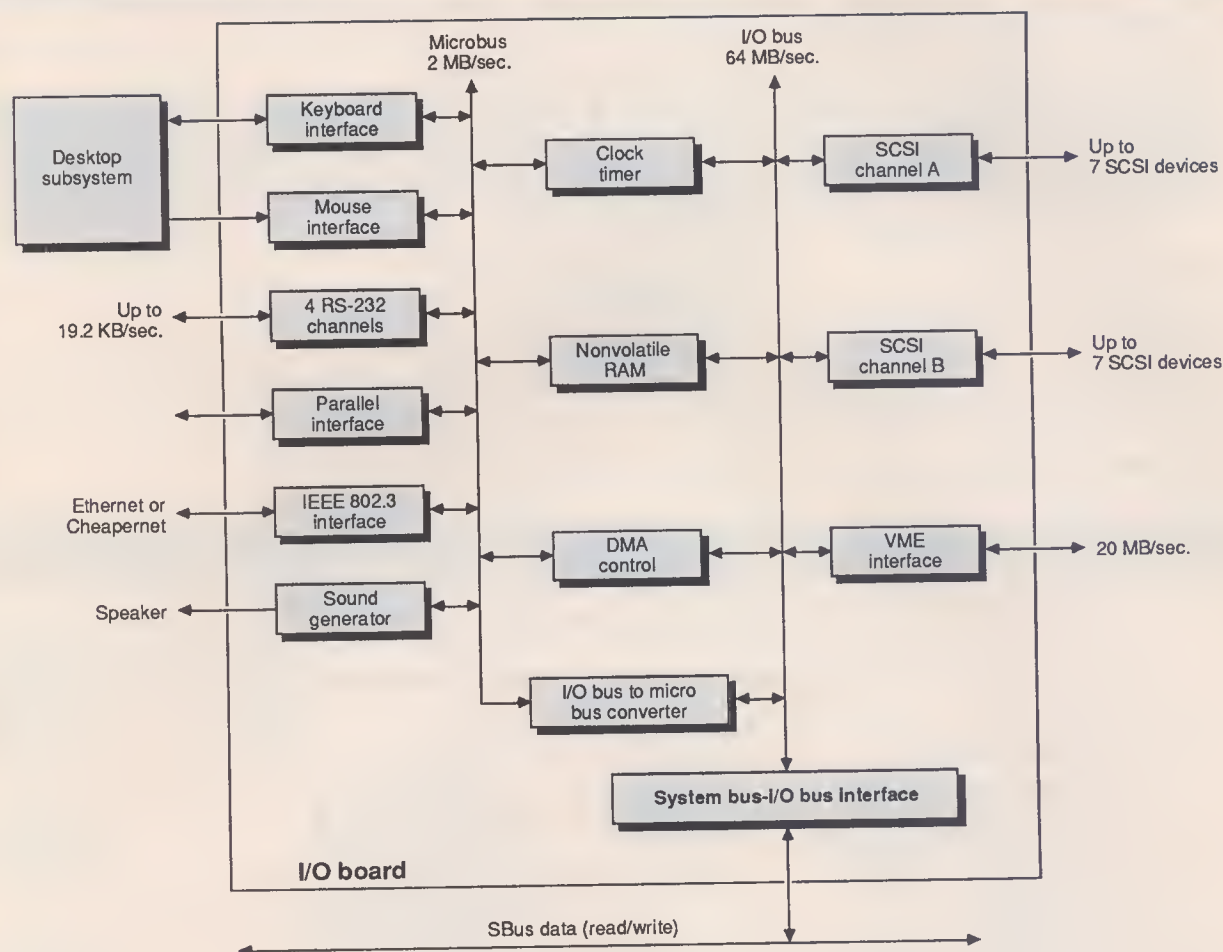


Figure 10. I/O subsystem block diagram.

for rotational position sensing controllers that can minimize effective rotational latency. Moreover, drive manufacturers typically introduce ESDI versions of new drives three months before SCSI versions.

SCSI also had two advantages. The synchronous SCSI transfer option promised much higher transfer rates than ESDI. SCSI also provided connection to such devices as quarter-inch streaming tape drives, half-inch tape drives, and optical disk drives.

We selected SCSI as the primary disk interface in the belief that it will offer higher performance in the long term. At least one vendor has already announced a multiple-spindle disk that provides an impressive four-megabyte-per-second transfer rate and less than 20-millisecond

access times. SCSI also provided immediate benefits since it eliminated the need for a separate tape controller.

Because of the savings in board area and cost, we were able to add a second SCSI interface. In combination with such operating-system features as disk striping, two SCSI controllers can easily outperform a single ESDI channel.

Our selection of a network proved much easier. Ethernet is the dominant network for scientific computers. The fiber-distributed data interface, though perhaps more suitable for machines in Titan's performance class, remains a future technology. A third alternative, Hyperchannel, has a significant position in the supercomputer community and is an option on Titan's VMEbus.

Implementing the architecture. Implementing the I/O subsystem architecture was an exercise in design fundamentals: bandwidth, duty cycle, and latency. By carrying over the basic design of the system bus, the available 64-megabyte-per-second I/O transfer rate more than met the bandwidth requirement. A simple bus arbitration scheme provides 100-percent bus utilization and can provide up to 50 percent of the bus bandwidth to a single requester.

A heavily pipelined machine delivers high data rates, but often at the expense of long latency due to pipeline delays. Read operations, for example, are affected because the requester must wait while the request flows forward through the pipe and the required data flows back through

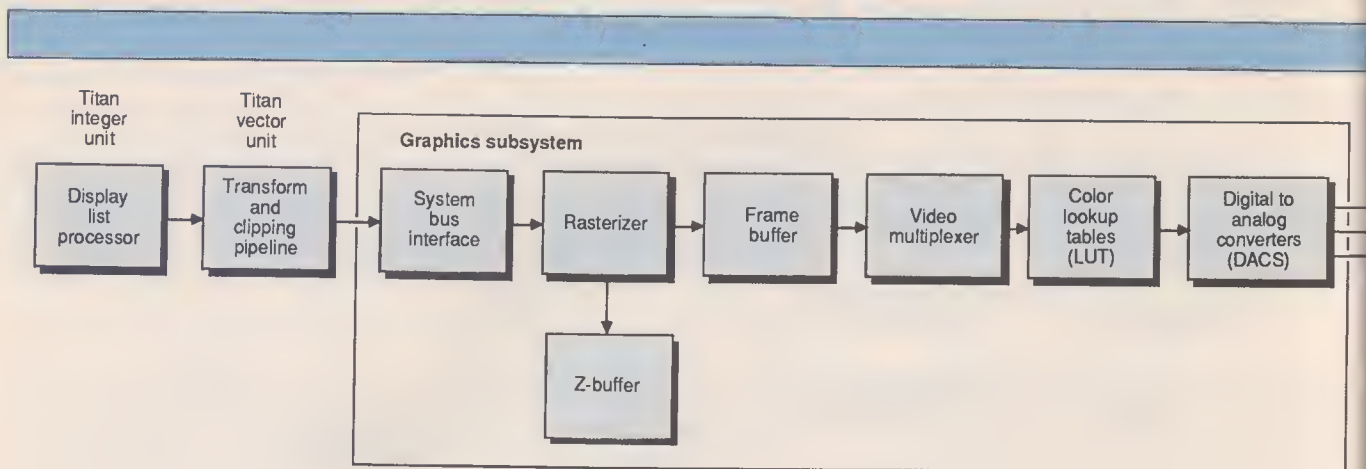


Figure 11. Graphics pipeline.

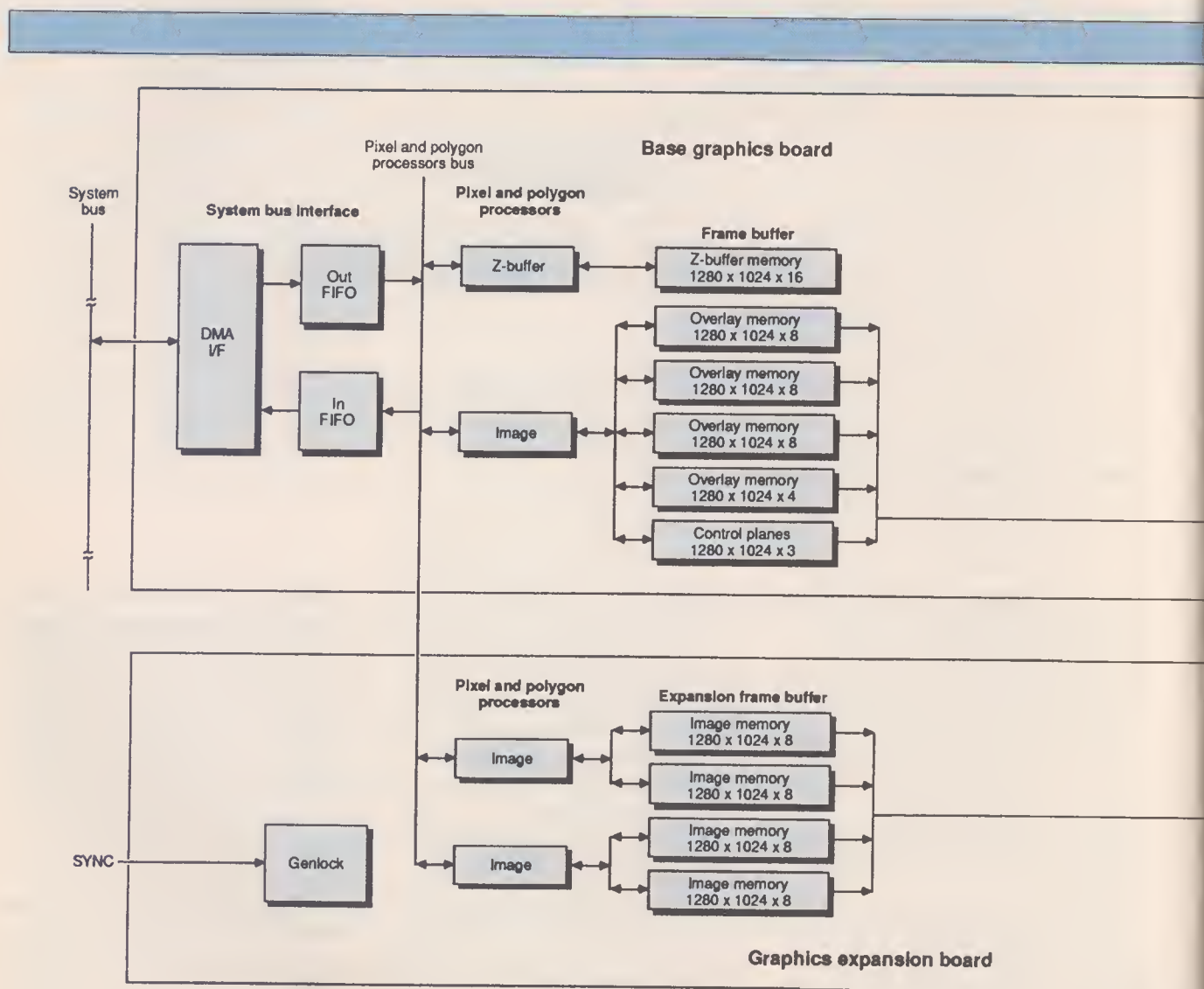
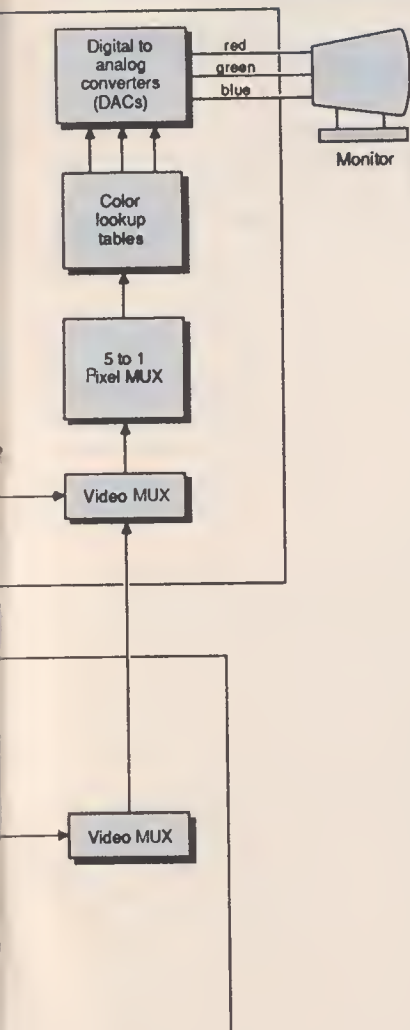
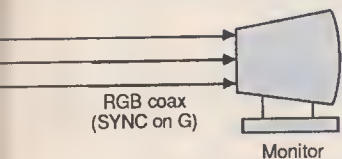


Figure 12. Graphics subsystem block diagram.



the return pipe. For I/O, this characteristic is exacerbated because I/O devices generally access memory at relatively low rates.

In aggregate, a large number of simultaneously active I/O channels make what amount to random memory accesses. Unless carefully managed, each access might be subject to the full pipeline delay. To avoid this problem, a buffered read-ahead scheme was used. Each DMA channel makes memory read requests in four-byte or eight-byte increments, regardless of the amount of data requested by the device. In addition, anticipating serial access, it also reads four, eight, or 16 bytes ahead.

The I/O bus is fast, but an interface to it is relatively expensive. Connecting each of the large-scale integration controller chips used in the I/O subsystem directly to the I/O bus would have required too much logic. Instead, the controller chips are connected via a slow-speed asynchronous bus called a microbus. A single-bus conversion circuit is used to connect the microbus to the I/O bus. A microbus is used as the command and control bus for the I/O subsystem and as the data path for low-speed devices.

Graphics subsystem

Titan's goal of visualization demands a graphics subsystem that provides high-quality dynamic images to the user in a multiported window environment. High-quality images, in turn, require 3D representation, high spatial resolution, high frame rates, and a large color palette. Furthermore, fine-grained decomposition of images into drawing primitives is needed for good shading quality. Dynamic images require at least 10 to 24 frames per second. They also require double buffering or some other method of switching cleanly between sequential frames.

Meeting the goals. Spatial resolution on the Titan display is $1,280 \times 1,024$ pixels per frame, with each frame scanned 60 times per second. This results in a pixel rate of 108 megahertz. While higher resolution and higher frame-rate monitors are becoming available, their cost and uncertain availability make them future technology.

The contrast ratios available on CRTs require over 100 shades per color for smooth shading, suggesting the use of eight bits of resolution per primary color

(red, green, and blue). Current technology (for example, Ramdacs from Brooktree) provides eight-bit digital-to-analog conversion at 125 megahertz, or better. Furthermore, this technology provides eight-bit converters, color lookup tables, and pixel multiplexing all in the same compact package. The Ramdac technology also provides a palette of 16 million colors, allowing ample resolution for smooth shading of almost any color.

Titan's spatial resolution and color resolution requirements pushed the size of the frame buffer memory to approximately eight megabytes. The decreasing cost of memory makes this frame-buffer size feasible. But bandwidth requirements for updating the display and generating images required special attention. The bandwidth for updating the display is addressed by dynamic video RAM technology.¹⁵ These memory chips are dual ported, where one port updates the display and leaves the second conventional parallel port available with a more than 90-percent utilization factor. The bandwidth for the image-generation process falls into three categories:

- (1) 3D transformation, clipping, and perspective divide.
- (2) Lighting calculations for the drawing primitives.
- (3) Pixel processing (scan conversion and bilinear interpolation).

The high performance of Titan's computing engine made it feasible to execute the transformations, clipping, lighting, and perspective-divide calculations on the integer and vector units. This approach enabled us to reduce the amount of special hardware dedicated to high-speed parallel pixel processors, and to provide sufficient frame-buffer memory for double-buffered, full-color images. Figure 11 shows how the graphics process is pipelined.

There are several advantages to this division of graphics computation:

- With higher level graphical operations executed in a general computing environment with high-level language support, there is a much more flexible growth path for the high-level graphical operations than is available in most microcoded or fixed-pipeline systems.

- Titan's dedicated hardware can focus on functions that require speed but not flexibility. These include lower level graphical functions such as scan conversion and interpolation. A block diagram of the lower level graphics functions are shown in Figure 12.

Table 4. Titan statistical summary.

Number of processors	1-4
MIPS per processor (peak)	16
MIPS per processor (avg.)	10
Physical address size	1 Gbyte
Virtual address size	2.0 Gbyte
Vector unit pipe rate	8 MHz
Floating-point rate (peak)	16 Mflops
Floating-point rate (avg.)	6 Mflops
Number memory pipes	2 Load, 1 Store
Number operation units	3
Load latency	8
Store latency	2
Operation latency	6
$n\frac{1}{2}(v = s * v + v)$	18
$r_{\infty}(v = s * v + v)$ uniprocessor	14.5 Mflops
Livermore Loops	1.7 Mflops
(Length 167 HM, uniprocessor)	
No. of integer registers	32
No. of vector-registers sets	8
No. of vector-registers/sets	32
Vector-register length	32
Bus width	16 Bytes
Bus bandwidth	256 Mbyte/sec.
Memory size	8-128 Mbyte
Memory access time (double)	8 cycles
Memory interleaving	8 or 16-way
I/O disk channels	2, 4 Mbyte/sec.
I/O VME bandwidth	15 Mbyte/sec.
Screen resolution	1280 × 1024
Pixel resolution	24 bit full-color
Screen refresh rate	60 Hz
Gouraud shading rate	200K 3D triangles/sec.
Vector drawing rate	400K 3D vectors/sec.
Memory/screen block transfers	5.2 million pixels/sec.
Screen/screen block transfers	17 million pixels/sec.

• Cost, size, and power requirements are all minimized.

Our pixel processing goal was to have a sustained pixel drawing rate of at least 15 to 20 million pixels per second for 100 pixel triangles. This requires a peak rate of about 50 million pixels per second for large drawing primitives. Given that the DVRAMs have a 256 × 256 organization, 20 chips per four bits of depth are needed to cover the display (five chips horizontally by four chips vertically). The five horizontal chips are interlaced in order to shift five pixels out of the serial shift registers at one time. This organization means that five pixels can be written or read at the same time on the parallel port.

The memory cycle for writing these five pixels takes about 400 nanoseconds. Five

pixels every 400 nanoseconds provides 12.5 million pixels per second or one quarter of the required rate. With four groups of DVRAMs interleaved in the vertical direction, it is possible to write each group of DVRAMs at the same time. This configuration provides sufficient DVRAM bandwidth to achieve the 50-million-pixels-per-second peak rate.

The organization of the pixel processors is essentially dictated by the DVRAM organization. Four separate pixel processors are used to drive the four vertically interlaced groups of DVRAMs. Hence, the writing and reading of pixels is executed four scan lines at a time. The concept of parallel pixel processors is then extended one step further, by using separate processors for each color channel and

one for the depth, or Z, buffer.¹⁶

To provide 3D representation, the hidden-surface problem is solved in low-level hardware. The technique involves storing 16 bits of depth information at each pixel location. The value stored at each pixel reflects the depth, or Z, position of the current pixel data. The depth value for new pixels is compared to the current depth value stored in the depth buffer. If the new value is closer to the viewer than the old value, then the new pixel is written; otherwise, it is ignored.

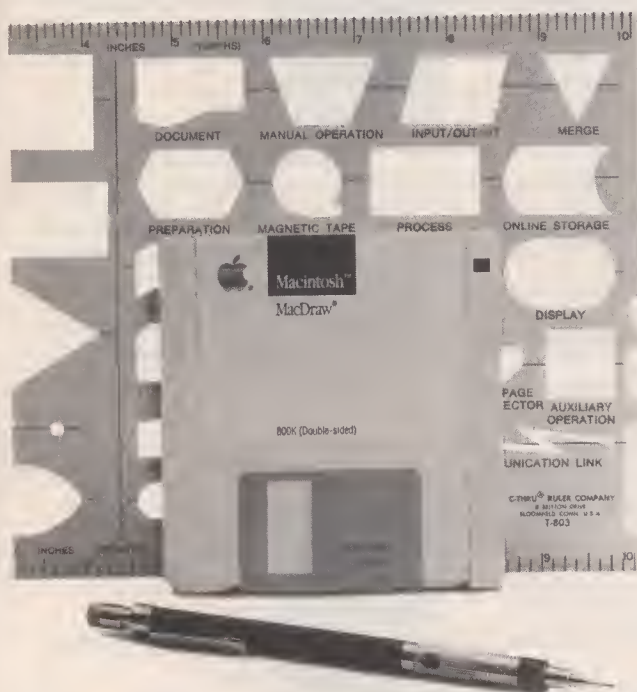
A key design goal here was to ensure that depth buffer operations did not impact the drawing speed of the pixel processors. That impact could be considerable since 3D depth-buffered operations require a read-modify-write sequence, unlike 2D operations which require only write. By building the depth buffer with DVRAMs, and using the serial shift register port to read the depth values one line at a time, the graphics subsystem does depth buffering with less than a five-percent decrease in pixel drawing rate.

Super- and minisupercomputers can provide the computational power required by many current scientific and engineering simulation, synthesis, and analysis problems. These machines do not have visualization capabilities and, thus, are often augmented by graphics workstations. A pair of machines, one used as a computational resource and one used as a visualization tool, fundamentally limits the way in which the engineer or scientist can work interactively. Graphics supercomputers, such as Titan, represent another evolutionary step in the development line.

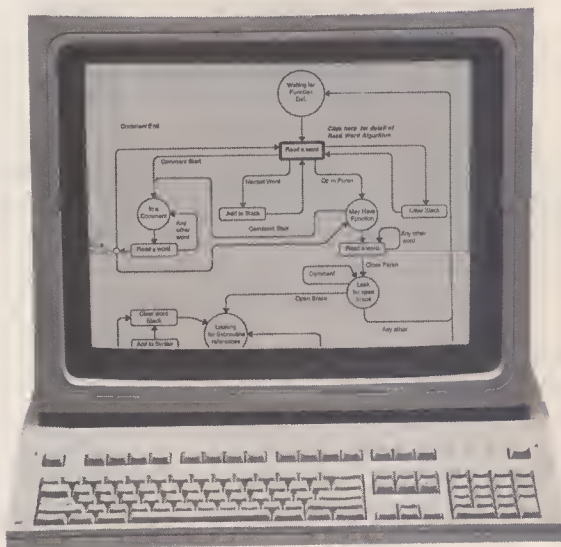
The architectural techniques used in Titan's design enable it to perform computationally like a minisupercomputer yet, at the same time, take an evolutionary step in development. Titan is like its minisupercomputer and supercomputer predecessors in its widespread use of parallelism, multiple processors, co-executing IPU and vector units, multiple pipelines within the vector unit, and concurrent processing of elements of vector instructions.

All of these techniques are aimed at reducing the effects of memory latency (caches, memory interleaving, pipelined requests) and concurrent processing of a large number of instructions. These uses of parallelism enable Titan to deliver a substantial amount of both integer and floating-point computational power: a two-processor Titan provides about one-

If you just need a drawing tool, use one of these



If you need a design tool, use Design/2.0™



A lot of tools can help you draw, but how many help you design and keep track of complex system models?

Design/2.0 is the powerful text and graphics processor that allows you to quickly create and edit flow charts, system models, technical documentation, organizational charts, computer programs, communication networks, business plans, and more. And Design/2.0 does it faster and more efficiently than any other tool!

End time consuming redrawing

Design/2.0 understands when objects in a diagram are related. Once you connect two objects, they stay connected. Move or resize an object and its connections are automatically redrawn. Design/2.0 also redraws all associated objects and text.

Use Design/2.0's flow chart symbols or create or import your own. Create and edit text anywhere in your diagram. Even do word searches. Associate text with objects or connectors. Or create hypertext links across pages.

The big picture is easy to see

As your model evolves, move detail to subpages. Build diagrams containing hundreds of hierarchically linked pages. Design/2.0 maintains all relationships and displays the hierarchy no matter how complex your model becomes.

Why not spend your time designing, rather than drawing?

Yes, I want to spend my time designing! Please send me more information on Design/2.0.



Name _____

Company _____

Address _____

City _____

State _____

Zip _____

Mail to: Meta Software, 150 CambridgePark Drive, Cambridge, MA 02140.

Or call 1-800-227-4106. Inside Massachusetts 617-576-6920

IECO 9/88

Meta Software

Available for the IBM® PC-AT, PS/2 & close compatibles. (Requires Windows™) \$350
Available for the Macintosh™ Plus, SE, II. \$250. Color available for the Mac II.

IBM PC-AT and PS/2 are registered trademarks of International Business Machines Corporation. Windows is a trademark of Microsoft Corporation. Macintosh is a trademark of Apple Computer, Inc. Design/2.0 is a trademark of Meta Software Corporation.

sixth the computational power of a uniprocessor Cray X-MP (as measured by the Livermore Loops or Linpack benchmarks) at one one-hundredth the cost.

Titan differs from super- and minisupercomputers in two important respects. The most interesting is the integration of graphics within the architecture. This provides a platform that facilitates implementation of applications that provide truly interactive computing and visualizing environments as opposed to the inherent batch feel of the present compute/transmit/display facilities provided by graphics terminals and computing servers, as shown in Table 4.

Graphics supercomputers also differ in that they are relatively inexpensive. At about \$10,000 per Mflops, they deliver about an order of magnitude better cost performance than any other kind of machine. At \$80,000, they can be used by a single user. □

Acknowledgments

Thanks to Cathy Miranker, Lizabeth Reilly, and Paul Ausick for their efforts in the production of this article.

References

1. J.J. Dongarra and I.S. Duff, *Advanced Architecture Computers*, ANL/MCS-TM-57, Argonne National Lab, Oct. 1985.
2. C. Rowen, et al. "RISC VLSI Design for System-Level Performance," *VLSI Systems Design*, Vol. 7, No. 3, Mar. 1986, pp. 81-88.
3. *Performance Brief*, MIPS Computer Systems, Issue 2.1, Mar. 29, 1987.
4. L.M. Censier and P. Feautrier, "A New Solution to the Coherence Problems in Multiprocessor-Cache Systems," *IEEE Trans. Computers*, Vol. C-27, No. 12, Dec. 1978, pp. 1,012-1,018.
5. T. Cheung and J.E. Smith, "A Simulation Study of the Cray X-MP Memory System," *IEEE Trans. Computers*, Vol. C-35, No. 7, July 1986, pp. 494-505.
6. J.E. Thornton, *Design of a Computer—The Control Data 6600*, Scott, Foresman and Co., Glenview, Ill., 1970.
7. D.W. Anderson, F.J. Sparacio, and F.M. Tomasulo, "The IBM System/360 Model 91: Machine Philosophy and Instruction Handling," *IBM J.*, Vol. 11, Jan. 1967, pp. 8-24.
8. J.R. Allen and K. Kennedy, *PFC: A Program to Convert Fortran to Parallel Form*, MASC Tech. Rep. 82-6, Dept. of Math. Sciences, Rice Univ., Mar. 1982.
9. W. Abu-Sufah, D. Kuck, and D. Lawrie, "Automatic Program Transformations for

Virtual Memory Computers," *Proc. 1979 National Computer Conf.*, June 1979.

10. D. Gannon and W. Jalaby, "Strategies for Cache and Local Memory Management by Global Program Transformation," *Proc. 1987 Int'l Conf. Supercomputing*, Springer-Verlag, June 1987, pp. 229-254.
11. J. Sanguinetti, "Micro-Analysis of the Titan's Operation Pipe," to appear in *Proc. 1988 Int'l Conf. Supercomputing*, Nov. 1988, pp. 190-196.
12. R.W. Hockney and C.R. Jessope, *Parallel Computers: Architecture, Programming, and Algorithms*, Adams Hilger, Bristol, England, 1981.
13. O.M. Lubeck, *Supercomputer Performance: The Theory, Practice, and Results*, LA-11204-MS, Los Alamos National Lab, Jan. 1988.
14. I.Y. Bucher, *Vector Performance of Register-to-Register Vector Computers*, LA-11198-MS, Los Alamos National Lab, Jan. 1988.
15. J. Nicoud, "Video RAMs: Structure and Applications," *IEEE Micro*, Vol. 8, No. 1, Feb. 1988, pp. 8-27.
16. J.W. Torborg, "A Parallel-Processor Architecture for Graphics Arithmetic Operations," *Computer Graphics*, Vol. 21, No. 4, *Proc. SIGGRAPH*, 1984, pp. 197-204.



Glen S. Miranker is chief architect at Ardent Computer. He previously served as engineering director at Valid Logic Systems in San Jose, Calif., and was involved in processor development at the IBM T.J. Watson Research Center in Yorktown Heights, NY.

Miranker holds a BS degree from Yale University and a PhD in computer science from the Massachusetts Institute of Technology.



Jonathan J. Rubinstein is manager of CPU design at Ardent Computer. Previously with Hewlett-Packard, he was the architect of the HP9000 Series 300 computer family.

Rubinstein earned BS and ME degrees in electrical engineering from Cornell University and an MS in computer science from Colorado State University.



Tom Diede is manager of graphics hardware at Ardent Computer. Previously, he worked in the same capacity at Edge Computer in Minneapolis, Minn., and as a designer of high resolution film recording devices at Dicom Corp.

Diede received his BSEE from the University of North Dakota.



Carl F. Hagenmaier is manager of I/O design at Ardent Computer. Previously, he held various positions with Burroughs at the Large Systems Plant in Mission Viejo, Calif., the Austin Research Center in Texas, and the Business Information Systems Group in Flemington, NJ.

Hagenmaier earned BS and MS degrees in electrical engineering from Stanford University.



William S. Worley, Jr. is vice president of systems architecture at Ardent Computer. Previously, he led development of the Spectrum Architecture and system software at Hewlett-Packard and taught computer science at the Illinois Institute of Technology and Cornell University.

Worley holds an MS in physics and information science from the University of Chicago and a PhD in computer science from Cornell University.

Readers may write to the authors at Ardent Computer, 880 W. Maude Ave., Sunnyvale, CA 94086.

National Center for Excellence in METALWORKING TECHNOLOGY (NCEMT)

The National Center for Excellence in Metalworking Technology is a technology transfer facility established in Johnstown, PA, which operates under contract to the U.S. Navy by Metalworking Technology, Inc., a non-profit subsidiary of the University of Pittsburgh Trust.

The mission of NCEMT is to provide a national center for the compilation, development, and dissemination of information on advanced metalworking technologies, ultimately providing a means for applying this technology to the solution of technical and productivity problems in the domestic metalworking industry.

NCEMT solves the problems of the metalworking industry through:

- On-line knowledge bases
- Process development
- Equipment demonstration
- Seminars and workshops
- Training and education
- Research and development

Immediate openings exist for the following:

Director of Computer and Information Systems

A Director of Computer and Information Systems with 6+ years of relevant experience who can provide the technical and managerial leadership to support a variety of networked applications using equipment from several vendors. Areas include: DECnet, TCP/IP, UNIX, VMS, artificial intelligence and expert systems, network management, automation tasks, FEM, CAD/CAM/CIM, SQL, and large database information access. A Ph.D. is preferred in Computer Science or Computer Engineering (or equivalent verifiable experience). Annual salary is commensurate with experience and qualifications. The Director will be responsible for managing the network environment and for the on-going development of an advanced information and management system.

Technical Reference Librarian

A Technical Reference Librarian who establishes and maintains the NCEMT technical library, responds to information requests, and conducts on-line searches. Position requires a MLS, 5 to 8 years' (minimum) experience in library administration, technical processing and reference, and comprehensive skills in conducting computer-assisted literature searches: DIALOG, BRS, etc. Candidate should possess a B.S. in Engineering, Science, or a related area, or an Advanced Certificate in Science-Technology Librarianship preferred. A strong interest in science and technology is a prerequisite, along with exceptional initiative, organizational ability and interpersonal skills.

Systems Specialist

A Systems Specialist to manage computer and network operations, vendor negotiations, and staff the Department of Computer and Information Systems. Position supports technical experts in workstation, PC, and mainframe applications, and provides for business and government reporting. Ph.D. is preferred in Computer Engineering/Computer Science (or equivalent related experience). The successful candidate also must have 6+ years' experience in a related position to include applications development with SQL, UNIX, VMS. Candidate should possess expertise in computer systems configuration, network management, electronic publications, office automation and systems support, and exhibit strong interpersonal skills. Software engineering skills and familiarity with business engineering applications are preferable. The position is highly visible and requires keeping abreast of state-of-the-art advances in computing and networking to assure rapid technology transfer.

NCEMT is located in the beautiful Laurel Highlands area of Pennsylvania, 80 miles east of Pittsburgh, and is near the University of Pittsburgh at Johnstown. Clean air, exceptional opportunities for outdoor activities, and a very low cost-of-living are among the area's many attractions.

Please send resume and salary requirements/history to:
Ms. Patricia A. Rylke, Human Resources Coordinator
Suite 206, East Hills Professional Building
1450 Scalp Avenue
Johnstown, PA 15904
Phone: (814)266-5395

High-Speed Computers of the Soviet Union

Peter Wolcott and Seymour E. Goodman

University of Arizona

In a speech to the Communist Youth League (Komsomol) in April 1987, Soviet Communist Party General Secretary Mikhail Gorbachev called supercomputer development a "top-priority task for our science and economy," essential for controlling processes in major industries, planning and modeling the economy, mastering space, executing defense tasks, and solving major research problems in science and technology.¹ He lauded the developers who worked to push computers to process one billion instructions per second by 1990 and 10 GIPS five years later. Gorbachev's emphasis on supercomputing underscores a wave of interest, research, and support in high-speed scientific/engineering computing that has grown steadily over the last decade.

Gorbachev's concern is well placed. For many years quantitative and qualitative deficiencies in computing have greatly hampered Soviet research and development in fluid dynamics, structural analysis, nuclear physics, weapons design, control systems, and other fields.

Gorbachev and many of his supporters also view scientific/technical progress as a cornerstone of *perestroika*, a program of economic modernization. In addition to providing solutions to many technical

**In the last decade,
Soviet development of
high-speed/advanced-
architecture
computers has
significantly
increased in scope,
profile, and support.**

problems, high-speed computers would be important high-profile standard-bearers of Soviet technical progress.

The USSR Academy of Sciences has always played a prominent role in the development of high-speed computers, in part because past work on such machines required less extensive production facilities but more intensive design research and development than other computers. As

Western experience has shown, this field presents a particularly fertile area for research, especially with regard to multiprocessor systems, because of the range of possible architectural approaches and the theoretical questions that must be addressed. The academy has embraced a number of multiprocessor approaches within the last decade, seeing the possibility of developing high-speed computers without necessarily using ultrasophisticated microelectronic components.

The academy also views supercomputing as a means to shore up an eroded position in computer development vis-à-vis the industrial ministries. Although the leading designer of computers through most of the 1960s, the academy achieved only moderate success in meeting the country's computing needs. The industrial ministries then undertook the large-scale development of functional duplicates of successful Western minicomputer and mainframe systems during the late 1960s and 1970s.² High-speed computers represent one of the most promising niches in which the academy can contribute significantly and recover its former status. In 1983 the academy formed the Department of Informatics, Computer Technology, and Automation (OIVTA), headed by academy

vice president Yevgeniy P. Velikhov, and charged it with supercomputer development as one of its top priorities.

Given the research-rich nature of high-speed computer development, we are not surprised that many of the academy's projects exhibit a greater degree of indigenous design than do those of the industrial computer-related ministries. This article gives an overview of these projects, from the mid-1960s to the present, and concludes with a discussion of the state of scientific computing and the impact of recent political and organizational changes.

BESM-6

S.A. Lebedev, the father of Soviet high-speed computing, began work in the late 1940s on the first Soviet electronic digital computer, a stored-program machine called the MESM (a Russian acronym for "small electronic calculating machine"). During the 1950s and early 1960s, as director of the Academy of Science's Institute of Precision Mechanics and Computer Technology (ITMVT) in Moscow, he developed several computers of the BESM series (an acronym for "large electronic calculating machine") with processing speeds of 7,168 to 20,480 operations per second (7-20 KOPS).

By 1964 Lebedev had developed the crowning work of his career, the BESM-6, which brought the Soviets almost up to world performance and sophistication levels for second-generation machines. This model features significant architectural advances such as 15 index registers, scratch-pad memory, multiprogramming with memory protection, and associative registers for use with virtual memory. Roughly 200 machines were manufactured from 1965 to 1977 at the Calculation Machines Plant (SAM) in Moscow. Although built with second-generation, transistor technology, this machine constituted the backbone of Soviet scientific computing into the early 1980s, and many are still in use.

The BESM-6 is a single-address, 50-bit-word machine with 32 Kwords of ferrite core main memory. It runs at a reported maximum rate of one million short arithmetic instructions per second.³ (The Soviets generally specify speeds in operations per second, although they often mean "instructions per second." They rarely give ratings in millions of floating-point operations per second.)

Although the BESM-6 shares some design features with the British Atlas and the CDC-1604, it does not closely copy either. It uses 24-bit instructions and stores two instructions to a word, like the CDC-1604, giving the BESM-6 a limited look-ahead capability. Since each instruction uses only 15 bits for addressing, no program can contain more than 32 Kwords. The BESM-6 has 46 basic operations and 14 definable opcodes to meet the needs of individual installations.

The BESM-6 employs virtual memory to allow multiprogramming, not to provide a larger programming space. Some later BESM-6 configurations have 64 or even 128 Kwords of main memory, but this improvement came much later when the size of the index registers increased. The 32-Kword maximum for arrays continued.

To implement virtual memory, the BESM-6 uses a pure paging system much like that of the Atlas but with larger, 1,024-word pages to reduce page faults. Main memory holds only 32 pages, putting a severe practical limit on the degree of multiprogramming. Eventually, operating systems provided multitasking of up to 16 processes simultaneously, although the tolerable limit is much less.

The BESM-6 can include up to 16 30-Kword drums and 32 996-Kword tapes, accessible by eight I/O channels that can transmit at an aggregate rate of 160,000 (50-bit) words per second. These are sharp improvements over earlier Soviet peripherals and reasonable achievements by then-world standards. The Soviets always have had more difficulty producing delicate electro-mechanical devices than electronic ones, and magnetic disks have been one of the weakest areas of Soviet computer technology; none were ever made specifically for the BESM-6.

The BESM-6 was eventually outfitted to access the 29-megabyte disks of the ES ("Unified System") mainframes—functional duplicates of the IBM System 360/370²—in place of drums. These disks, first produced in the mid-1970s, transmit data at 156 Kbytes per second. We do not know to what extent the BESM-6 has been used with 100-megabyte disks. This deficiency in transmission rates and in reliable, high-volume secondary storage prevents BESM-6 users from performing the computations on large amounts of data typically needed in supercomputer applications.

A basic set of systems software was to be available by the time the BESM-6 went into production. But the BESM-6 hard-

ware proved more difficult to program than originally supposed; the first operating system developed for the prototype was inadequate, and a usable operating system was not developed until two years after production started. For many years the burden of developing significant systems software rested on the users themselves. To meet their own needs, two of the first recipients of a BESM-6—the Joint Institute of Nuclear Research (OIYal) in Dubna and the Institute of Applied Mathematics (IPM) in Moscow—began developing operating systems and compilers, and built a significant portion of BESM-6 systems software over the next decade.

The Dubna and Dispak operating systems have dominated BESM-6 use. The Dubna operating system, developed at OIYal, is a batch processing system that can run a foreground and a background process concurrently. A significant number of BESM-6 users built their applications on Dubna, and it remained in regular use into the 1980s.

At the same time as the OIYal efforts, the IPM was developing the forerunner of Dispak, a batch processing system that supported some unknown number of concurrently executing processes. It probably did not incorporate any virtual memory capabilities.

Although Dispak was completed by 1971, its modular design made significant modifications possible. The system underwent regular development and improvement into the 1980s, acquiring interactive facilities, remote access capabilities, translators for nearly two dozen languages, and the ability to manage clustered BESM-6s linked to common disks and drums.

The Soviets did not produce another publicly announced large-scale computer until the late 1970s. Many rumors circulated about several other BESM projects, including one to match the CDC-6600, but no known BESM-6 successor went into production before Lebedev died in 1974. The next ITMVT computer to be publicly described, the EI'brus, was first delivered no earlier than 1979. Lebedev's record of development had been regular and respectable, and he had had both viable ideas and experienced personnel. So why did such a gap in development exist?

The answer is not clear. The developers probably tried to implement overly ambitious designs that could not be supported by the quality of components and development tools available. There also may have been serious design flaws not detected until

Table 1. A parameter comparison of the El'brus, B6700, and B7700 computers.

	El'brus	B6700	B7700
Date of announcement	1978	1970	1970
Date of first delivery	1979	1971	1973
	198? (El'brus-2)		
No. CPU (max)	10	3	4
No. I/O processors (max)	4	3	7
Total No. processing units in configuration (max)	14	6	8
Word size	64-bit	48-bit	48-bit
Processing speeds (El'brus-1)			
Adding fixed-pt. Nos.	520 ns	200 ns	130 ns
Adding floating-pt. Nos.	780 ns	200 ns	130 ns
Multiplying half-word Nos.	780 ns		
Multiplying full-word Nos.	1,300 ns	2,000 ns	800 ns
Memory module size	144/288 Kbytes (16/32 Kwords)	384 Kbytes (64 Kwords)	768 Kbytes (128 Kwords)
Min memory increment	4 modules	1 module	1 module
Max main memory*	32 modules (4/8 Mbytes, 512/1024 Kwords)	16 modules (6 Mbytes, 1 Mwords)	8 modules (6 Mbytes, 1 Mwords)
Memory access time	500 ns per 9 bytes	770 ns per 6 bytes	1.5 μ s per 24 bytes
I/O processor data-exchange rate	15 Mbytes/s	1.6 Mbytes/s	6 Mbytes/s
Data transmission processor:			
Internal storage (max)	16 Kwords	16 Kwords	16 Kwords
No. DTP in configuration (max)	16	8	8
No. of communications lines in configuration (max)	4,096	2,048	2,048

*Sources differ. Some clearly state a maximum main memory of 512Kwords; others 1024K.

later. When V.S. Burtsev succeeded Lebedev as ITMVT director in 1973, there also was a delay as he consolidated his position and reoriented the institute toward the development of an entirely new type of machine.

In use for more than two decades, BESM-6s on the average have probably logged more hours of operation per unit than any other computer model in the world. Before 1979, no other general-purpose processing systems with speeds greater than a few hundred KOPS were available in more than trivial quantities. Many prominent military, industrial, and academic facilities have used BESM-6s extensively. Each facility developed a cadre of technicians intimately familiar with the BESM-6, making it a well-maintained machine in a country infamous for its terrible computer service.²

The BESM-6 also has had one of the largest and most active user groups in Soviet history. Thousands of users have programmed on the machine and many have learned to do so quite well, developing and improving software even after the BESM-6 was taken out of production. By 1977 over a million instructions of systems software had been written, plus many more in applications software. In the following ten years, additional monitor systems, editors, debuggers, and compilers were developed.

However, the BESM-6 has hampered Soviet efforts to do world-class research requiring intensive computing. Inadequate operating systems and a small programming space have forced most programmers to code in low-level languages, including machine code, even though the first high-level translators were developed

within a few years of the hardware. The BESM-6's limited disk capacity and reliability produce continual headaches, and the slow development of basic tools has forced users to develop their own or do without. Users at each installation often customized the functions of certain definable opcodes and developed their own compilers and compiler versions, making it difficult to share applications software.

Remarkably, the BESM-6 still supports a significant portion of Soviet scientific computing. An entire section of a 1986 computer conference in Kiev was devoted to interfacing the BESM-6 with peripherals and other computers.⁴

El'brus

When Burtsev became the new director of ITMVT, he initiated the development of a family of multiprocessors patterned after the established Burroughs architecture. The first El'brus-1 was installed in 1979 and the El'brus-2, in the mid-1980s. We do not know how many El'brus-1s have been installed; there are probably at least 10, but possibly as many as 50. Only three or four El'brus-2s are in operation.

El'brus architecture. The El'brus designers relied heavily on the stack-based, modular architecture of the Burroughs B6700 and B7700 to meet their design goal of a general-purpose machine with high throughput and reliability. Table 1 offers a parameter comparison between the El'brus-1 and the B6700 and B7700.^{5,6} Differences between the El'brus-1 and -2 lie chiefly in component composition and processing speeds, and unless otherwise noted, the figures apply to the El'brus-2 as well. The central processing units, main memory modules, and I/O processors are connected by central switchboards so that any CPU and any I/O processor can directly access any memory module. Special data transmission processors manage network functions.

The designers report processing rates of 1.5 million and 12 million instructions per second for the single-processor and 10-processor El'brus-1 versions respectively. The corresponding figures for the El'brus-2 are 12 MIPS and 125 MIPS. These figures exceed those reported by Burroughs for their machines only because they are based on continual execution of fixed-point arithmetic operations and do not consider time for memory access, I/O,

and more complex instructions.

Only minor logical differences exist between the stack-based El'brus CPU and B6700/B7700 processors. Like the B7700, the El'brus uses local memory buffers to store the data and instructions for a process. While the B7700 also employs an associative memory buffer to store frequently used operands and descriptors that will not fit in the stack buffer, the El'brus employs its associative buffer for storing data common to more than one process, simplifying operand management but occasionally reducing performance.

Main memory is interleaved in a set of between four and 32 modules. Unlike the Burroughs machines, which use a pure segmentation mechanism for memory management, the El'brus uses a combination of segmentation and paging. Paging allows noncontiguous storage of segments, although a segment brought into main memory is stored as a segment and referenced by a single descriptor in the process stack, as in the Burroughs machines.⁶

The El'brus designers made great efforts to provide the machine with adequate access to secondary storage. A single El'brus can have up to four I/O processors, each accommodating 24 peripheral device control units and up to four data transmission processors. Like the BESM-6, the El'brus uses ES peripherals and is similarly constrained by the state of disk and tape production in the Soviet Union. An El'brus today probably uses 100- to 200-megabyte disks with data transfer rates of 806 Kbytes per second and tapes with rates of 738 Kbytes per second. Since data can be transferred simultaneously from four disk and 16 tape drives, the maximum transfer rate per I/O processor is nearly 15 megabytes per second.

A considerable effort to develop El'brus systems software paralleled hardware development efforts. The native, high-level programming language of the El'brus, El'-76, was developed in parallel with the hardware to provide convenient high-level programming structures, efficient execution, and job control functions in a single language. While Burroughs provided a spectrum of languages to meet these needs, the El'brus designers have strongly promoted the idea that one language can be used for all needs.

El'-76 is based on Algol-68, whose block structure easily executes on a stack-based architecture. By including external object types, El'-76 extends the principles of interprocedure data protection to interprocess protection, making it well suited

El'brus designers have strongly promoted the idea that one language can be used for all needs.

as a systems programming and job control language.

Conceptually, El'-76 differs little from Algol-68. Typing and operator definition are less flexible, but condition-handling structures are better developed. While both Algol-68 and El'-76 include parallel-programming constructs, El'-76 provides parallel execution only at the procedure level. This construct is better suited to the El'brus environment, where the basic unit of parallelization is the process.

While the El'brus operating system has much the same design and functionality as the Burroughs Master Control Program—the operating system of the 700 series—it is not compatible. We have too few details about the operating system to determine how similar it is to the Burroughs product or how well it functions.

The SVS-1 processor. From the outset of El'brus development, the scientific community clearly was not eager to convert to the machine. Since it had an entirely different architecture from the BESM-6, a complete transition would require an overhaul of many man-years' worth of BESM-6 software. Since the El'brus contained many more hardware modules and connecting switches that could (and did) break down frequently, maintenance became a significant consideration. BESM-6 users were reluctant to trade a working BESM-6 and many experienced technical repairmen for a rather unreliable El'brus with few who could repair it expeditiously.

Designers at the ITMVT softened the transition by developing the SVS-1, a second type of El'brus CPU compatible with BESM-6 software and El'brus hardware. It enables BESM-6 users to use all of their old software in an environment with significantly expanded main and peripheral

memory. While the SVS-1 uses integrated-circuit technology, has a modified arithmetic unit, and includes means of connecting it to the El'brus multiprocessor configuration, it resembles the BESM-6 processor enough to be maintained by existing BESM-6 repair crews. It is two to three times faster than the BESM-6 processor.

The El'brus vector processor. The El'brus vector processor can be a processor in an El'brus-2 configuration. In design and performance, this processor is the most Cray-like of any of the Soviet machines.⁷ As an add-on to the El'brus, it has a scaled-down design and slower components relative to the Cray-1. It has a 40-nanosecond cycle time (versus 12.5 nanoseconds for the Cray-1) but uses five pipelined vector arithmetic units with chaining to achieve a theoretical peak processing rate of 125 million floating-point operations per second, or 80 percent the speed of the first Cray-1. The processor's eight memory modules have a total capacity of 524,288 64-bit words. This processor has been under development for several years, but we do not know how far development has progressed nor do we know of any operational units.

El'brus status. On paper, the El'brus computers represent a significant achievement for the Soviets. But the El'brus has experienced serious production problems and delivery postponements. For example, the Institute of Cybernetics of the Estonian Academy of Sciences (IKANESSR), a leading computer research and development institution, was approved in 1978 for an El'brus delivery in 1979-1980, but didn't actually receive a working model until 1986.

Not enough El'bruses—even those with a sole SVS-1 processor—are available to replace all the BESM-6s. The teleprocessing capabilities of the El'brus, which allow more than one facility to access the machine, have not been used effectively.

Some installations have managed to keep their BESM-6s running. A few have foreign machines like the CDC-6500, but a lack of spare parts makes these difficult to maintain. The only practical alternative has been to use higher-end ES mainframes, possibly with attached array processors. By the mid-1980s, a few hundred data processing mainframes running 1 MIPS to 5 MIPS were available, some of which could accommodate up to 10 array processors. But array processors have not

been produced in corresponding quantities.

Potential users have become very cynical about ITMVT and the EI'brus. Even when the machines are delivered, many users continue to rely on their ES mainframes. An employee of one organization that acquired an EI'brus a couple of years ago said, "We haven't decided what we want to do with it yet." Their EI'brus mostly lies idle.

Decision makers within the academy also became impatient with the lack of progress. Burtsev was fired around 1985 and replaced by G.G. Ryabov, a longtime ITMVT engineer. Ryabov has vigorously promoted ITMVT efforts in the media and refocused nearly all resources on the EI'brus-3, a 16-processor machine with a projected processing rate of one GIPS. A prototype is promised by 1990 and a 10-GIPS successor by 1995.

M-10

As Burtsev developed the EI'brus machines, M.A. Kartsev, another engineer with many years of experience with high-speed computers, developed the M-10, a 32-bit synchronous multiprocessor system that was apparently operational in 1979. It has eight 16-bit processors that together can process data at a reported rate of 5 MIPS.⁸ Although Kartsev had ties with the academy, he was most closely associated with the military-industrial sector.

The M-10 is distinguished by its ability to conveniently parallel-process data of different formats, dynamically changing the clustering of processors to match the format of the data at hand. Data can be fetched from memory in units of between two and 64 bytes, and the eight 16-bit processors can execute identical instructions on two 64-bit, four 32-bit, or eight 16-bit numbers.

A few M-10s are used to model seismic activity and plasma behavior, but the project's status is uncertain. Kartsev died in 1983 without a clear successor to carry on his work. Little evidence exists of continuing production or research on the machine.

MARS

The late 1970s and early 1980s witnessed a considerable broadening of Soviet large-scale and advanced-architecture projects.

MARS-T addresses some fifth-generation issues, but some Soviet scientists prefer to call it a "new-generation" machine.

The ITMVT remained most visible, but other organizations took advantage of the abundant research opportunities in multiprocessor systems and began projects of their own.

The Modular, Asynchronous, Expandable System is an umbrella name for two projects currently under the direction of the Siberian Division of the USSR Academy of Sciences (SOANSSSR) in Novosibirsk. The projects, MARS-M and MARS-T, grew out of work begun in the mid-1970s on hierarchical, open architectures with asynchronously communicating modules. The chief designer is V.Ye. Kotov, deputy director of the Computer Center at SOANSSSR.

Kotov and others convinced the State Committee on Science and Technology (GKNT) to establish a temporary working group called Start, which in 1985 received a three-year mandate to prototype MARS-M, a number-crunching multiprocessor for scientific computation. The developers soon realized that their work had broader application and began to develop a prototype "fifth-generation" multiprocessor called MARS-T with a flexible architecture for diverse applications and distributed configurations, object-oriented data handling for knowledge bases, and easy-to-use interfaces for nonprogrammers. Single-processor versions will serve as engineering workstations. Because MARS-T addresses some fifth-generation issues, the press and key Start scientists have labeled it a fifth-generation computer. However, it does not have nearly the raw processing speed (10 GIPS) that Japanese developers feel is necessary for fifth-generation machines. Some Soviet scientists, therefore, prefer to call MARS-T a "new-generation" computer.

Start is the Soviet Union's first experiment with a new organization type, the Temporary Interbranch Scientific Complex (VMNTK), which pools research

projects that best address specific problems. The Computer Center of the SOANSSSR is the lead organization, responsible for developing the hardware and systems software. Its partners are the Institute of Cybernetics of the Estonian Academy of Sciences (IKANESSR) in Tallinn, the Computer Center of the USSR Academy of Sciences (VTsANSSSR) in Moscow, and the Scientific Research Institute of Control Computers (NIIUVM) in Severodonesk, which is under both the academy and the Ministry of Instrument Construction, Means of Automation, and Control Systems (Minpribor).

MARS-M. MARS-M was developed to process scalar values and short arrays as efficiently as large arrays and to prototype the principles of asynchronous communication.⁹ It contains four asynchronously communicating subsystems: three pipeline processors for address calculation, computation, and control, and a common memory store. Processors communicate through channels implemented as queues in the common, two-million-word, interleaved memory. Each processor contains functional units that communicate through a high-speed crossbar switch. Processors and functional units both have individual stores of local memory plus access to common memory at their respective levels through which they share data. By pipelining instructions through functional units that operate in parallel, the MARS-M can run at up to 20 Mflops.

MARS-T. The MARS-T design is based on a hierarchy of functional modules to provide parallel processing while limiting the complexity of control. MARS-T implements functional modules differently at different levels in the hierarchy: as individual processors, as software modules, or as functional units within a processor. All functional modules can store and process data, have local control, and transfer data to other modules. Modules at each level can execute in parallel.

Niklaus Wirth's Lilith/Modula-2 personal workstation and Inmos' transputer multiprocessor have clearly influenced the MARS-T designers. To combine an object-oriented approach with underlying parallel processing, the Soviets have developed a processor called Kronos with characteristics of both the transputer and the AM 2901-based Lilith processor (see Table 2).¹⁰⁻¹²

Like the Inmos transputer, MARS-T supports concurrent processes within the

instruction set of its processors. Like the Lilith processor, Kronos uses an instruction set that executes Modula-2 efficiently, providing a firm footing for a sophisticated user interface, data abstraction, and information hiding through modules. First produced in 1986, the 32-bit Kronos differs from the Lilith processor primarily in its increased word length, simplified instruction fetch, and inclusion of asynchronous transputer communications functions in the instruction set.

MARS-T treats programs as a set of fundamentally independent and parallel processes and subprocesses to be executed by the different functional subsystems. Sequentiality is viewed as the exception. Execution can be data driven, event driven, or condition driven.

MARS-T software. A number of software development projects have pursued the goal of a powerful, easy-to-use, intelligent workstation for engineers. These projects are developing parallel processing languages, object-oriented programming environments, tools for knowledge-base development and use, and data manipulation tools, all geared toward the engineer without extensive computer training. Some of the most capable programmers in the Soviet Union are working on MARS-T software, with some respectable results.

To meet the widest number of programming needs, two basic operating systems have been developed, each with a kernel that manages the execution of programs on multiple processors: Unix (including its associated programming tools) for C programmers, and Excelsior, a more efficient system that provides Unix-like functionality to the user but is oriented towards Modula-2. On top of these run the user applications programs and the object-oriented programming environments, such as Nut.

Nut (for New Utopist) is an object-oriented language currently under development at the IKANESSR by E. Kh. Tyugu and others. It is characterized by a high level of data abstraction, structural synthesis of programs, and knowledge-based programming.¹³ By incorporating many facets of knowledge representation and inference, Nut contains the tools necessary for building small expert systems. It is primarily used for defining a problem domain (typically a domain for engineering applications) and enabling a user to draw inferences and make calculations about that domain. Nut operates in an impressively fast, easy-to-use windows

Table 2. A parameter comparison of the Kronos, T424 transputer, and Lilith processors.

	T424 transputer (1984)	MARS Kronos (1986)	Lilith AM 2901- based processor (1979)
Size	32-bit	32-bit	16-bit
Main RAM memory	4 Kbytes ultra high speed 4 Gbytes addressable	512 Kbytes- 2 Mbytes	64 Kbytes (original) 8 Mbytes addressable (current)
Cycle time	50 ns	250 ns	150 ns
Processing speed	10 MIPS	2-3 MIPS	3 MIPS
Instruction set	64 instructions, all single byte	256 instructions, mostly single byte	256 instructions, mostly single byte
Architecture	Stack	Stack	Stack
Used in multi- processing systems?	Yes	Yes	No
Inter-processor communications	Self-synchronizing channels, common memory modules	Asynchronous channels, memory modules	N/A
No. high-speed serial links between processors	4	Variable	N/A
Speed of links	1.5 Mbytes/s	1-4 Mbytes/s	N/A
Assembly language	Occam-like	M-code extended	M-code
Basic HOL	Occam	Modula-2	Modula-2

environment developed at the IKANESSR on 32-bit Australian Labtam computers.

Nut differs from other object-oriented languages in the use of program synthesis and the inclusion of production rules to represent certain types of knowledge. Essentially, the program synthesizer takes a description of a problem domain and, if possible, synthesizes a program to generate a specified set of output values for a specified set of input values. This practice has a firm theoretical foundation, but is limited by the need for a clearly understood and specified problem domain.

Nut, the corresponding windows environment, and a set of knowledge-base development tools constitute the software to make MARS-T an engineering CAD workstation. The IKANESSR actively uses the windows environment. Nut is in the final testing stages, and the knowledge-base development tools, incorporating a hybrid rules/frames/semantic-net approach to knowledge representation, have been implemented on a sequential

processor. The attention to powerful, easy-to-use software distinguishes MARS-T from other Soviet advanced-architecture projects and indicates a desire to make widespread use of the machine.

MARS status. Start was disbanded as anticipated in April 1988, having successfully produced prototypes of MARS-M and MARS-T.¹⁴ The MARS-T prototype currently has eight processors. The organization of future research and development is still being debated, but Start's successor will likely rely in part on newly established private cooperatives for maintenance services and software development.

ES-1766

The ES-1766 is a macro-pipeline multi-processor that pipeline-processes data across processors. Different processors repeatedly execute different portions of an algorithm and send their output to their

neighbors. The designers report that the ES-1766 is theoretically capable of 100 MIPS.¹⁵

The theoretical foundations were laid during the mid-1970s by V.M. Glushkov, director of the Institute of Cybernetics of the Ukrainian Academy of Sciences (IKANUKSSR) in Kiev. A prototype was developed by 1984, and series production reportedly began in 1985.

We know little about the machine's architecture and the relationship, if any, between the ES-1766 and the ES series of mainframes. Better documentation exists for the software development environment, however. The foundation is MAYaK (a Russian acronym for "macro-pipeline language" that translates as "lighthouse" or "beacon"), a family of integrated programming languages geared for use by different classes of users that allows sophisticated programmers to specify how to execute the pipelining.¹⁶

MAYaK's top level consists of a structured sequential language for applications programming that hides all hardware details from the user. Parallelization can be specified only through the coding of modules external to each other; nested modules must be executed sequentially. The operating system manages the assignment of routines to processors. The other three languages give the programmer progressively more control over the definition of system subsystems, the communication between them, and their execution on logical processors.

One of the strongest aspects of the ES-1766 project has been the systems development environment. The ES-1766 was designed with a computer-aided design system called PROYeKT, which includes tools not only for hardware design, but also for the simulation of program execution. With these tools a programmer can run MAYaK programs on a single-processor mainframe and monitor their performance while varying the number of processors, their speed, the number and throughput of interprocessor channels, and the distribution of tasks. In this manner the optimal algorithms and hardware configuration can be determined. PROYeKT will prove a significant asset to ES-1766 users since the power of large-scale systems is largely determined by the quality of the software.

PS-x000

Several projects in the late 1970s aimed to fill a serious gap in high-speed scientific

Running at a reported peak processing rate of 200 MIPS, the PS-2000 is currently the fastest Soviet computer in series production.

and real-time computers for specialized applications such as geophysics, fluid dynamics, and process control. A joint effort of the Institute of Control Problems (IPU) and the Scientific Research Institute of Control Computers (NIIUVM) produced the PS-2000 and PS-3000 multiprocessors. Although they have certain features in common, the machines have significantly different architectures and belong to two different development lines. Running at a reported peak processing rate of 200 MIPS, the PS-2000 is currently the fastest Soviet computer in series production. It has been in production since about 1983 and is used mainly in geophysical applications.

PS-2000. I.V. Prangishvili and others at IPU began design work on the PS-2000 in the mid-1970s.¹⁷ This machine consists of a parallel processor with between 8 and 64 processing elements (in eight-element blocks) together with a monitor subsystem and an external memory system. In the parallel processing system, the elements are linked together under a single control unit that broadcasts control signals and 24-bit operators for simultaneous execution on all or some processing elements. Each element can store up to 16,384 24-bit words of data received either from the control unit or directly from external memory.

The machine uses a flexible set of channels for data exchange. A parallel bus links each element with its two neighbors, and a serial bus links all processing elements, but can be segmented to cluster them into groups of 8, 16, or 64, so data that is local to a few processors can be shared. A third channel allows direct memory exchange between the control unit, external memory system, and element memories.

Extensive use of multiple types of registers and local processors to distribute (and speed up) the execution of various

functions and reduce multitasking overhead distinguishes the parallel processor. The control unit, for example, contains 80 registers of more than ten different types. Multiple registers for accessing local memory allow single-cycle execution of multioperand instructions.

One of the main drawbacks of the design is the 24-bit word length. Only four bits are used for an exponent, limiting the range of numbers that can be represented.

Effective use of the PS-2000's parallelism is difficult. Some high-level language translators take advantage of the parallelism, but they have not received adequate distribution and, more significantly, do not produce efficient code. The vast majority of the software developed for the PS-2000 has been written in microcode or mnemokod, an assembly language. Here the programmer may be overwhelmed. The instruction set contains instructions sufficient for controlling all aspects of computation and dataflow within the system, but properly using this capability requires detailed knowledge of the system's architecture. As a result, generic, multipurpose code is difficult to write. Although several libraries of standard routines have been developed over the last five years, many of the routines are extremely specialized, performing such functions as "add two vectors of length 1,024 or 4,096."

The IPU has not played an adequate role in providing systems and applications software. A PS-2000 operating system is reportedly available, but the continuing need for applications programmers to control communications functions with the host and peripherals indicates otherwise.¹⁷ Individual installations must develop their own applications.

PS-3000. IPU designed the 32-bit PS-3000 multiprocessor with provisions for scalar and vector processing, sophisticated interrupt handling, and 256 megabytes of virtual memory. Used primarily for real-time process control, it has one control processor linked with three computation processors in a single-instruction, multiple-data fashion. Two or possibly four such configurations can be tightly coupled to form an multiple-instruction, multiple-data machine. Each computation processor has three arithmetic units that work in parallel at a rate of 10 to 20 MIPS on vector data.¹⁸

Although production of the PS-3000 reportedly began in 1984, we know little about the quality and availability of its

software. Parallelizing compilers are being developed, and a reasonably powerful, easy-to-use operating system is at least in the test phase.

There have been reports of a PS-4000, a successor to the PS-3000 incorporating some artificial-intelligence capabilities, but no details about the architecture or state of development are available.

TAGREI programmable architecture multiprocessor

For the last decade, A.V. Kalyayev of the Taganrog Radio Electronics Institute (TAGREI) in Taganrog has led efforts to develop a multiprocessor computer with a programmable architecture. Although this machine has received much less publicity in the Soviet Union than have the El'brus and MARS computers, six of the largest scientific research institutes of the Ministry of the Electronics Industry (Minelektronprom) are developing the crucial integrated circuits and switching mechanisms. The goal is a 256-processor system especially suited for solving systems of differential equations. The fact that this project is based on an unproven design approach—a programmable architecture—makes it unique among the major Soviet large-scale systems efforts.

The TAGREI dataflow machine consists of a set of processors, each comprised of an elementary processor, a 14-nanosecond microswitch, a local elementary operation memory, and an elementary switching memory.¹⁹ The user can program the links between processors to produce a desired processor configuration: pipeline, hierarchical, array, etc.

The basic processing and switching commands are macros whose definitions are stored locally in each processor. This enables the specification of processing functions within processors while allowing processors to interact with each other in a data-driven manner. Data can pass between processors along data channels or via common memory.

The TAGREI machine processor sends data directly to another processor, rather than to a logical node that is then mapped to a physical processor. This might limit the number of nodes in this dataflow machine; the designers gave up a measure of generality and flexibility in favor of speed.

The architecture can function as a pipeline processor, an array computer, or a

The TAGREI project is based on an unproven design approach, making it unique among the major Soviet large-scale systems efforts.

number of other architectures. Architecture programming is quite difficult, though; the programmer must specify macroinstructions at or close to a microcode level. Once defined, however, the macroinstructions provide the user with a nearly high-level language that translates quite directly to executable code. The literature to date has not given any examples of what these macroinstructions look like.

The TAGREI project shows a typically long development cycle. Although Kalyayev started designing the machine over ten years ago, some of the fundamental components such as the hardware switches reached the prototype stage only three or four years ago.

Nevertheless, in a significant break with tradition, the GKNT and Minelektronprom have strongly supported this unconventional architecture. Kalyayev is now a corresponding member of the Department of Informatics, Computer Technology, and Automation (OIVTA) and was recently named the director of a new organization, the Scientific-Research Institute for Multiprocessor Computer Systems (NIIMVS).

IPM multiprocessor

Over the last several years, a group of computer designers at the Institute of Applied Mathematics (IPM) in Moscow has worked out a design for a 128×128 -processor computer for fluid dynamics problems.²⁰ The target processing speed is one to 10 Gflops.

The IPM developers have proposed an MIMD architecture with a design approach similar to that of the Flow Model Processor proposed by Lundstrom and Barnes for NASA. One program is written, but a copy is distributed to each processor, which executes it indepen-

dently. Synchronization need take place only at certain points in the execution. The 16,384-processor grid steps sequentially through 128 plains to process a calculation point space of more than $2 \cdot 10^6$ points.

The IPM model uses a much larger number of processors than the FMP (16,384 versus 512) and subsequently relies on simpler processor-to-processor and processor-to-common-memory connections. Because the IPM model has enough processors to cover the calculation points needed for one calculation point plain in the applications envisioned by the designers (a 100×100 grid), its synchronization constructs are simpler than those of the FMP.

Each processor consists of a 64-bit data processing unit, a 32-Kword local memory store, and eight input and eight output buffers for receiving and sending data to neighboring processors. One control processor and a common memory module connect to 128 5-Mbytes-per-second buses, each connected to 128 processors. The buffers are controlled by an interface unit that not only regulates the flow of data into and out of the buffers, but also is able to mask certain buffers to further regulate the exchange.

According to the designers, the paper version of this computer has been completed. If the chief obstacle to constructing a prototype is the lack of enough 64-bit 1-Mflops microprocessors, then building a slower, 32- or even 16-bit version simply to test the design would be reasonable. Although available literature gives no indication that this has been done, it would be surprising if the design had not passed the paper stage. G.I. Marchuk, the president of the Academy of Sciences, has given the design high-level publicity recently, indicating renewed efforts to involve industry and implement the design.

Other machines

Some gaps remain in our knowledge of Soviet high-speed computer systems. For example, Western visitors to the Institute of Space Research (IKI) were told of the IZOT 1703, a multiprocessor capable of 120 Mflops, jointly developed by IKI and the Bulgarian Central Institute of Computer Technology.²¹ This is the first evidence of international joint programs to develop high-speed computers. The extent of such programs remains unclear. Since 1974 there have been scattered references to work at the Institute of Cybernetics in

Kiev on recursive architecture machines that dynamically reconfigure themselves for different tasks. A prototype might currently exist, but the most recent literature remains largely theoretical.²²

Rumors of high-speed "secret machines" intended for military-related applications have also existed through the years. While such classified efforts certainly do exist, it is unlikely they are producing spectacularly better results than the projects we know about. Two former "secret machines," the M-10 and the Vesna, an unsuccessful predecessor of the BESM-6 running at 200-300 KOPS, have not proved in retrospect to be extraordinary technological achievements by then-world standards.

In the last decade, Soviet development of high-speed/advanced-architecture computers has significantly increased in scope, profile, and support. With a variety of architectures and some organizational changes, the Academy of Sciences is seeking to develop the computing base to support the scientific-technical progress deemed critical to the national economy and defense, and at the same time reestablish its own prominence in Soviet computer development.

For over two decades the Soviet scientific community has been caught between an aging collection of inadequate second-generation machines and a new generation of high-speed computers that for the most part are still promises. Whatever the precise fates of rumored successors to the BESM-6, none were produced in substantial numbers. The El'brus-1 and El'brus-2 were anticipated as the new backbone of scientific computing, but the program failed in the eyes of the thousands of scientists without access to the few machines that became operational. Other serially produced high-speed systems have not become widely available.

Since the academy lacks production facilities, it has relied on the industrial ministries to manufacture the computers it designs. Although the ministries and the academy have cooperated better in high-speed computer manufacturing than in other areas, this division of labor has resulted in serious production problems. Mismatches between specifications and the components available to support them have crippled production. The academy has a history of designing systems that cannot be easily manufactured in quantity; often the ministries must rework large por-

Leadership at the highest levels recognizes the scientific, economic, and military need for high-speed computers.

tions of a design before they can produce the machine.

Ministries such as Minradioprom also depend on the chemical and electronics ministries for crucial base materials and components. Continuing problems with mass production and poor interministry relations frequently result in components too poor in quality to support the academy's designs adequately. Furthermore, the academy does not have the authority to dictate to industry what to produce.

As the need for large-scale and very large scale integration components has grown, the academy and the computer production ministries have had difficulty in getting Minelektronprom to develop and produce chips of the required quality and quantity. The ITMVT's close working relationship with Minelektronprom to develop El'brus components has not proved sufficient, nor has it helped other academy organizations working on high-speed systems.

What are the prospects for Soviet scientific computing? Certainly the increased national and international support for high-speed computer development offers some hope that conditions will improve. Leadership at the highest levels recognizes the scientific, economic, and military need for high-speed computers as well as their importance as a high-profile symbol of technical progress.

Within the Academy of Sciences, the OIVTA should provide a more focused distribution of resources among developers and a greater integration of developers with their microelectronic support industries. Steps have also been taken to establish much-needed microelectronics support industries within the academy itself. A few years ago a scientific-technical association was established in Leningrad specifically to develop tools for VLSI chip

production. Additional organizations such as the Institute for Problems of Microelectronic Technology and Super-Pure Materials have been created to develop the materials to produce advanced components.

Perestroika and Gorbachev's personal attention are not by themselves likely to overcome production difficulties. However, they may have a greater impact here than in other sectors of the computer industry, since the infrastructure involved in high-speed computer development is not large.

For the near future, the prospects for significant increases in the quality and availability of scientific computers remain dim. While the newer machines described in this article exhibit some promise, it is too early to tell whether their actual performance will meet expectations.

Since the ITMVT is directing nearly all of its resources at the El'brus-3 and -4, it is unlikely that the El'brus-1 and -2 will be improved and produced in significantly greater quantities. Even assuming that a prototype El'brus-3 is developed on schedule in 1990, series production of more than a few units a year might not begin until the mid-1990s. In contrast, Cray Research has installed approximately 200 machines since 1976 and anticipates a 1989 production rate of one system per month for the Cray Y-MP, announced in early 1988.

Similar prospects face the ES-1766, MARS, and TAGREI efforts. The ES-1766 prototype was reportedly finished in 1984. Preparations were being made for series production in 1986, but a substantial inventory of basic systems software had yet to be completed.⁴ Significant production may not begin until the end of the decade, if then. The development of MARS and the TAGREI multiprocessor have not come even this far, and we doubt they will be widely available before the early to mid-1990s.

The scientific computing environment must also be improved. Scientists suffer not only from insufficient processing power, but also from unreliable large-volume disk stores, unreliable I/O peripherals, uneven electrical power grids, and even inadequate supplies of printer paper. Weak telecommunications and networking facilities leave many scientific institutions electronically isolated.

Gorbachev's promises of machines processing 1, 10, or even 100 GIPS notwithstanding, the Soviet scientist for some time to come will have to be content to use BESM-6s and ES mainframes, some of the

latter with attached array processors. Though also suffering production problems, newer general-purpose mainframes such as the ES-1066—which runs at 5 to 12 MIPS—are being delivered and will bear the bulk of Soviet large-scale scientific computing. □

Acknowledgments

We gratefully acknowledge the assistance and encouragement of Sidney Fernbach, Duncan H. Lawrie, John P. Riganati, and others on the IEEE Supercomputer Subcommittee. We also thank E.Kh. Tyugu and V.Ye. Kotov for informative conversations.

References

1. M.S. Gorbachev, Moscow Television Service in Russian at 5:31 a.m. Greenwich Mean Time on April 16, 1987. Live address by Mikhail Sergeyevich Gorbachev, General Secretary of the CPSU Central Committee, at the 20th Komsomol Congress in the Kremlin Palace of Congresses.
2. N.C. Davis and S.E. Goodman, "The Soviet Bloc's Unified System of Computers," *Computing Surveys*, Vol. 10, No. 2, June 1978, pp. 93-122.
3. W.B. Holland, "The BESM-6 Computer," *Soviet Cybernetics: Recent News Items*, No. 1, Feb. 1967, pp. 6-18.
4. A.A. Kalenchuk-Porkhanova and T.S. Yakovleva, "Hardware and Software of Complexes and Systems for Joining Non-homogenous Computers," *Upravlyayushchiye Sistemy i Mashiny [Control Systems and Machines]*, No. 2, Mar.-Apr. 1987, pp. 124-125.
5. "Burroughs B6700 and B7700," Dataproc Research, Delran, N.J., Jan.-Mar. 1977.
6. B.A. Golovkin, *Parallel Computer Systems*, Nauka, Moscow, 1980.
7. V.S. Burtsev et al., "A Vector Processor with a Programmable Structure," in *Computer Processes and Systems*, G.I. Marchuk et al., eds., Vol. 2, Nauka, Moscow, 1985, pp. 63-72.
8. M.A. Kartsev, "The M-10 Computer," *Doklady Akademii Nauk [Addresses to the Academy of Sciences]*, Vol. 245, Mar. 1979, pp. 309-312.
9. Yu. A. Vishnevsky et al., "A Modular Asynchronous Developed System," *Kibernetika [Cybernetics]*, No. 3, May-June 1984, pp. 22-29.
10. D.N. Kuznetsov et al., "The Kronos Processor in a Multiprocessor System," in *Computer Systems and Software*, V.Ye. Kotov et al., eds., Siberian Division of the Academy of Sciences of the USSR, Novosibirsk, 1986, pp. 13-19.

11. R. Ohran, "Lilith and Modula-2," *Byte*, Vol. 9, No. 8, Aug. 1984, pp. 181-192.
12. D. Pountain, "Microprocessor Design," *Byte*, Vol. 9, No. 8, Aug. 1984, pp. 361-366.
13. E.Kh. Tyugu et al., "Nut: An Object-Oriented Language," *Computers and Artificial Intelligence*, Vol. 5, No. 6, 1986, pp. 521-542.
14. "Computers of the Future are Being Built," *Sotsialisticheskaya Industriya [Socialist Industry]*, No. 113, May 15, 1988, p. 1.
15. V. Petrenko, *Pravda Ukrainy [Pravda Ukraine]*, Sept. 3, 1982, p. 2.
16. S.S. Gorokhovskiy et al., "The Algorithmic Language MAYAK," *Kibernetika [Cybernetics]*, No. 3, 1984, pp. 54-77.
17. I.V. Prangishvili et al., *Parallel Computer Systems with Common Control*, Energoatomizdat, Moscow, 1983.
18. V.V. Rezanov and V.M. Kostel'yanskiy, "Computer Technology Facilities for Controlling Technological Processes and Processing Geophysical Data: Status and Development Prospects (SM-1210 and PS-3000)," *Pribory i Sistemy Upravleniya [Control Instruments and Systems]*, No. 1, Jan. 1983, pp. 7-12.
19. A.V. Kalyayev, "Multiprocessor Super-systems with a Programmable Architecture, Based on the Dataflow Principle," in *Computer Processes and Systems*, G.I. Marchuk et al., eds., Vol. 2, Nauka, Moscow, 1985, pp. 140-153.
20. A.N. Andrianov et al., "On the Structure of the Computer for the Solution of Flow Problems: Systems Approach to Design," in *Computer Processes and Systems*, G.I. Marchuk et al., eds., Vol. 2, Nauka, Moscow, 1985, pp. 13-62.
21. "Supercomputer Used at Research Center," *Aviation Week and Space Technology*, Oct. 12, 1987, p. 27.
22. V.A. Myasnikov, M.B. Ignat'yev, and Yu.Ye. Sheynin, "Architecture of a Modular Multimicroprocessor Computer System," *Kibernetika [Cybernetics]*, May-Jun. 1984, pp. 48-53.



Peter Wolcott is a PhD student in the Department of Management Information Systems at the University of Arizona, Tucson. His research interests include software and large-scale scientific computing systems in the Soviet Union and Eastern Europe.

Wolcott received his bachelor's degree (magna cum laude) in computer science and Russian from Dartmouth College in 1984.



Seymour E. Goodman is a professor in the Department of Management Information Systems at the University of Arizona, Tucson. His research interests include international developments in computer technologies, with emphasis on the USSR and Eastern Europe, and related public policy issues.

Goodman received his bachelor's degree from Columbia University in 1965, and his PhD from the California Institute of Technology in 1970.

Readers may write to the authors at the MIS Dept., University of Arizona, Tucson, AZ 85721.

DID YOU KNOW?

- Foxboro's unprecedented growth based on the success of our Intelligent Automation Series has created significant opportunities for high-tech professionals.
- That our unique approach allows our technical professionals to move across functional areas, expanding their technological base in the areas of high-performance distributed process systems.
- Operating Systems (real-time UNIX™), Graphics, Communications (ISO), Real-Time Control, Man-Machine Interface are just some of the technologies you'll be exposed to at Foxboro.

If you're looking for a career with an industry leader that is constantly advancing the state-of-the-art and whose high-tech products touch the lives of people around the globe, come to Foxboro.

PRINCIPAL ENGINEERS

You will play a key role in the Phase 2, 32 bit implementation for our recently introduced Intelligent Automation Series. These are Technical Leadership positions and will involve defining, designing, coding and the integration of software.

An appropriate degree or equivalent with 5-7 plus years of progressive, hands-on software development with real-time microprocessor systems and "C" or assembly language level programming is required. Graphics, data base configuration management or control software background desirable.

Foxboro offers a complete benefits package including immediate eligibility for health and dental plans, 100% tuition assistance and 401K investment plan with matching company contributions.

There's much more you should know about Foxboro. To learn more call Dick Smallwood, Professional Staffing, at (508) 549-3991, or send your resume to his attention at The Foxboro Company, MS N05-1A, 38 Neponset Ave., Foxboro, MA 02035. Foxboro is an equal opportunity employer, M/F.



The Intelligent Automation People

FOXBORO

UNIX is a trademark of AT&T Bell Laboratory.

Fuzzy Concepts in Expert Systems

K.S. Leung and W. Lam
Chinese University of Hong Kong

Most of today's commercial expert-system building tools use certainty or confidence factors to handle uncertainties in the knowledge or data.¹ But they cannot cope with fuzzy concepts such as *tall*, *good*, or *hot*, which constitute a very significant part of a natural language. In fact, some of these fuzzy concepts have been incorporated into several expert systems, such as Cadiag-2² and Fault,³ which are purposely built from a high-level language for a specific domain of application. In Cadiag-2 the knowledge representation of fuzzy concepts is designed specifically for medical diagnosis. In Fault some fuzzy reasoning is also supported. Several AI programming languages, such as FProlog, also provide mechanisms to handle fuzzy concepts.⁴ FProlog is similar to Prolog except that in FProlog a truth value expressed numerically is allowed in a fact. The uncertainty can then be handled automatically by the FProlog interpreter.

This article presents a comprehensive expert-system building tool, called System Z-II, that can deal with exact, fuzzy (or inexact), and combined reasoning, allowing fuzzy and normal terms to be freely mixed in the rules and facts of an expert system. This fully implemented tool has been used to build several expert systems in the fields of student curriculum advisement, medical diagnosis, psychoanalysis, and risk analysis. System Z-II is a rule-

**The expert system
shell System Z-II
handles both exact
and inexact reasoning.
It allows any
combination of fuzzy
and normal terms and
uncertainties.**

based system that employs fuzzy logic and fuzzy numbers for its inexact reasoning. It uses two basic inexact concepts, fuzziness and uncertainty, which are distinct from each other in the system.

Inexact knowledge representation and reasoning

Much human knowledge is vague and imprecise.^{5,6} Human thinking and reason-

ing frequently involve inexact information. Expert systems should therefore be able to cope with such inexact information, from the following possible sources:

- inherent human fuzzy concepts,
- unreliable information,
- matching of similar rather than identical experiences,
- incomplete information, and
- differing (expert) opinions.

Types of inexact knowledge. In System Z-II four types of inexact information have been classified:

(1) Uncertainty occurs when one is not absolutely certain about a piece of information. The degree of certainty is usually represented by a numerical value. Example:

X is a bird. (0.8)

If X is a bird, then it can fly. (0.9)

The certainty factors are 0.8 and 0.9.

(2) Fuzziness occurs when the boundary of a piece of information is not clear-cut:

John is quite young.

If the price is high, then the profit should be good.

Quite young, *high*, and *good* are fuzzy terms.

(3) Uncertainty and fuzziness may occur simultaneously in some situations:

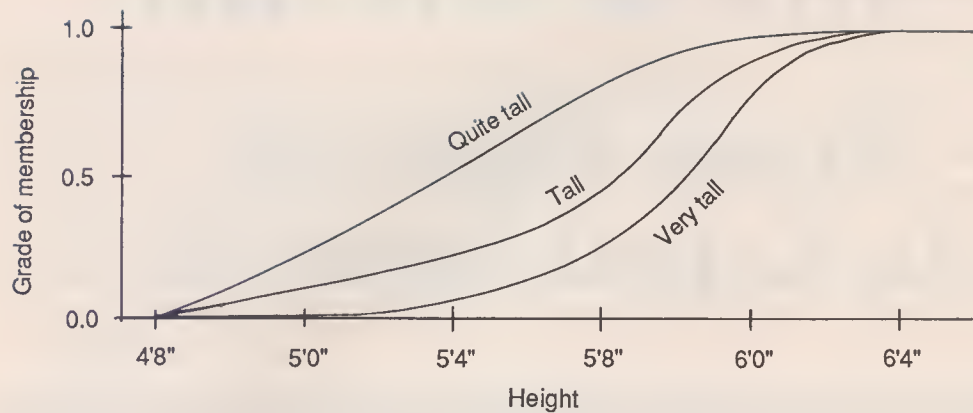


Figure 1. Fuzzy sets of fuzzy terms with modifiers.

John is rather tall. (0.8)

If the price is high, then the profit should be good. (0.9)

The certainty factors are 0.8 and 0.9; *rather tall*, *high*, and *good* are fuzzy terms.

(4) Sometimes the uncertainty can also be fuzzy:

John is very heavy. (around 0.7)

Here, *around 0.7* is the fuzzy uncertainty, and *very heavy* is a fuzzy term.

Dealing with inexact knowledge. The common approaches to inexact knowledge in expert systems⁶ are summarized in the following paragraphs.

Bayesian approach. Based on probability theory, the Bayesian approach can deal only with uncertainty. Collecting or estimating all the prior conditional and joint probabilities required for this method is difficult for domain experts. However, it has been suggested that employing conditional independence assumptions can reduce the number of probabilities to be estimated.⁶ This approach also depends on the availability of a complete set of hypotheses, and hence its applicability is restricted.⁵

Certainty factors. This approach can deal only with uncertainty. A certainty factor $CF(h, e)$ is a numerical value between zero and one that stands for the degree of confirmation of the hypothesis h based on the evidence e . Certainty factors are used

in the Mycin system to handle uncertainty in evidence (facts) and rules.⁵ For example:

rule:

IF X is a bird,
THEN it can fly. ($CF = 0.9$)

fact:

X is a bird. ($CF = 0.8$)

conclusion:

It can fly. ($CF = 0.9 * 0.8 = 0.72$)

One advantage of this approach over probability theory is that it does not require prior probabilities and therefore does not require a large volume of statistical data. Moreover, experts are more comfortable assigning certainty factors to the facts and rules.⁵ In fact, certainty factors have been widely adopted in expert system shells such as Emycin and S.I to handle uncertainty.

Dempster-Shafer theory of evidence. The Dempster-Shafer theory calculates belief functions—measurements of the degree of belief. The theory allows the decomposition of a set of evidence into separate, unrelated sets of evidence; a probability judgment can be separately assigned to each set of evidence.⁷ This approach, however, involves many numerical computations, and in the case of a long inference chain the structure of the resulting belief function would be very complex.

Fuzzy logic. Expert systems can use fuzzy logic to handle fuzzy concepts and approximate reasoning.⁸ For instance, the fuzzy term *tall* can be defined by the following fuzzy set:

Height	Grade of membership (possibility value)
4'4"	0.0
4'8"	0.0
5'0"	0.1
5'4"	0.2
5'8"	0.7
6'0"	0.9
6'4"	1.0
6'8"	1.0

These possibility values constitute a possibility distribution of the term *tall*. Modifiers such as *very*, *around*, and *rather* are common in approximate reasoning. We can obtain the possibility distribution of a fuzzy concept like *very tall* or *quite tall* by applying arithmetic operations on the fuzzy set of the basic fuzzy term *tall*. For example, we can calculate the possibility values of each height in the fuzzy set representing the fuzzy concept *very tall* by taking the square of the corresponding possibility values in the fuzzy set of *tall*, as shown in Figure 1.

The rule *If the price is high, then the profit is good*, with fuzzy concepts *high* and *good*, can be modeled by a fuzzy relation R . Let A_1 and A_2 be the fuzzy sets representing the concepts *high* and *good*, respectively. We obtain the fuzzy relation

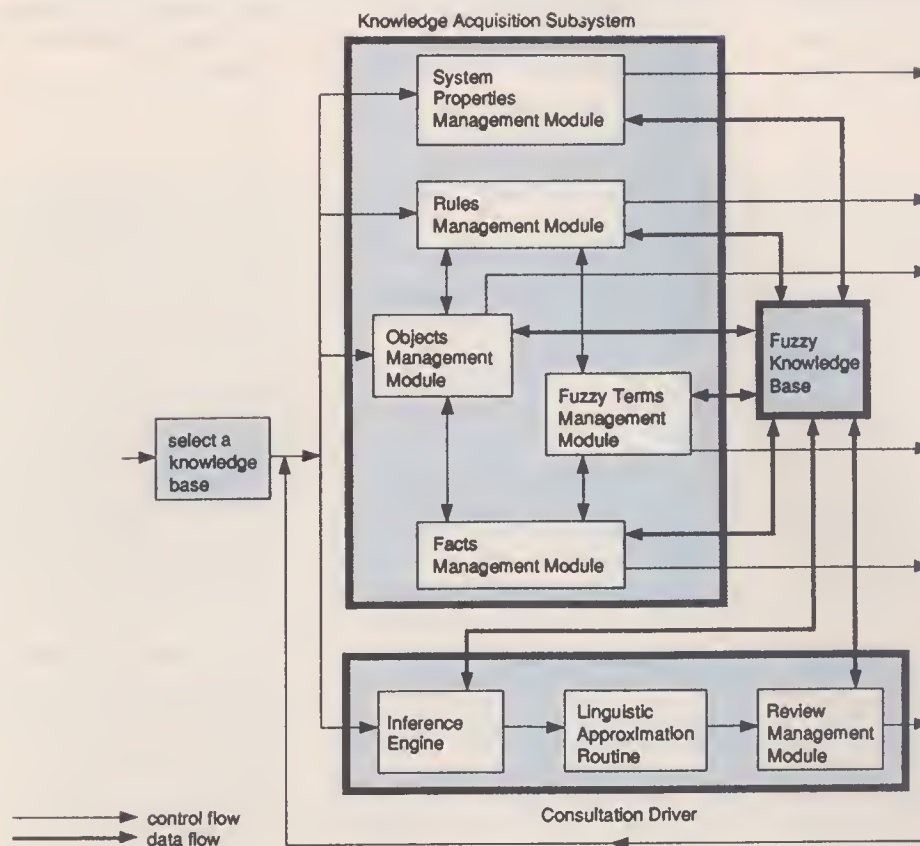


Figure 2. System Z-II's architecture.

R , represented by a matrix, by performing fuzzy operations on A_1 and A_2 , expressed as vectors. Many researchers⁸ have proposed methods of computing the fuzzy relation R ; some of them are presented later in this article in the "Consultation driver" section.

If a fact *The price is very high* matches the rule, the fuzzy concept *very high* can be represented by a fuzzy set F , which we obtain by applying an arithmetic operation (a square operation in this case) on A_1 . Then we compute the fuzzy set C , representing the fuzzy term in the conclusion, by applying a fuzzy operator called composition (denoted by $\circ$) on F and R : $C = F \circ R$.

The formula for the fuzzy composition is as follows:

composition: $C = F \circ R$

$$\mu_C(x) = \max_w (\min (\mu_F(w), \mu_R(w, x)))$$

where $\mu(x)$ is a membership function and w and x are elements in the universe of discourse.⁸

As a result, vector C will indicate *very good* and the conclusion *The profit is very good* will be drawn. This operation forms a simple inference from a fact and a rule both containing fuzzy terms.

System description

System Z-II is an expert system shell that facilitates the construction of rule-based consultation systems. Its major characteristic is that it allows any mix of fuzzy and normal terms as well as uncertainties in the rules and facts. To achieve this task, it employs fuzzy logic to handle inexact reasoning and fuzzy numbers to handle fuzzy uncertainty. Moreover, its menu-driven and restricted, natural language

interfaces based on fuzzy logic are specifically designed for easy knowledge engineering and consultations.

The development environment of Z-II consists of VAX-Lisp version 2.1 and VAX-Pascal version 3.4 running under VMS version 4.5 on a VAX-11/780 computer. In fact, Z-II is developed mainly in VAX-Lisp with part of its inference engine written in VAX-Pascal to substantially increase execution speed.⁹ VAX-Lisp is an implementation of Common Lisp, a highly portable, efficient, and powerful dialect.

Basically, System Z-II consists of three subsystems: the knowledge acquisition subsystem, the consultation driver, and the fuzzy knowledge base (see Figure 2). The components of the knowledge acquisition subsystem include management modules for objects, facts, fuzzy terms, rules, and system properties. These mod-

Table 1. Contents of predefined object slots.

Slot	Contents
TYPE	Indicates type of an object
FUZZY-OR-NOT	Indicates whether values of an object are fuzzy or not
ASK-FIRST-OR-NOT	Indicates whether value of this object is obtained by asking questions first or deducing from known facts and rules
USED-BY-RULES	List of rules whose antecedent parts contain this object
UPDATED-BY-RULES	List of rules whose consequent parts contain this object

ules are responsible for acquiring and managing rules and facts, which may contain any mix of fuzzy and normal terms and uncertainties. The task of the fuzzy knowledge base is to store all these knowledge entities.

The consultation driver consists of three modules: the inference engine, the linguistic approximation routine, and the review management module. The function of the inference engine is to extract the knowledge stored in the fuzzy knowledge base and to make inferences from the respective rules and facts. The linguistic approximation routine maps a set of fuzzy sets onto a set of linguistic expressions or descriptions, translating a fuzzy set into natural language after Z-II has drawn a conclusion. The review management module handles various reviews requested by users.

Knowledge acquisition subsystem

Objects management module. This module creates, modifies, or deletes objects in the system. An object is a basic entity in the system. It is uniquely identified by two elements: an object name and an attribute. For instance, the term *the weight of the body* is represented by the object BODY WEIGHT, with BODY being the object name and WEIGHT being the attribute. An attribute may be empty if the object name is sufficient to describe the object. An object is instantiated to a single value or multiple values during a consultation. It possesses a number of predefined slots that specify its properties. (The use of slots for knowledge representation is similar to the frames

approach.) The contents of slots should be given in the knowledge acquisition phase. The objects management module provides routines to manipulate objects as well as their slot contents. The contents of some of the slots are listed in Table 1.

Fuzzy terms management module. Normally, the values of an object are literal strings or numbers. However, if an object is fuzzy (indicated by the FUZZY-OR-NOT slot), its associated values can be fuzzy expressions such as *very tall* and *rather good*. These fuzzy terms are represented by fuzzy sets, and the fuzzy terms management module provides routines to define fuzzy sets for corresponding fuzzy terms. A fuzzy set is effectively a list of numbers. The management module uses Lisp mapping functions to manipulate lists and individual elements in a list, making it easy to implement the primitive fuzzy set operations.

Facts management module. System Z-II has two alternative ways to enter facts. One method is for the user to enter restricted English sentences. The other is invoked by the system asking for information about an object and its certainty step by step. The facts management module can either be invoked by the user or by the inference engine when it finds that some required facts are missing.

A fact is actually a data proposition of this form:

<OBJECT> is <VALUE> (fuzzy/nonfuzzy uncertainty)

The value of a numeric object is a number, while a nonnumeric object contains a string of symbols. If an object is fuzzy, however, its value is a linguistic expression

such as *very tall* or *quite good*. *Very tall* and *quite good* are represented by two fuzzy sets obtained by taking the square and square root of the fuzzy sets representing the basic fuzzy concepts *tall* and *good* respectively.

Fuzzy uncertainty is modeled by fuzzy numbers representing the concepts *around 0.8*, *close to 1.0*, and so on. A fuzzy number is actually a real-number fuzzy set that is both convex and normal. The definitions of convex and normal fuzzy sets are given in the sidebar on fuzzy numbers.

Fuzzy numbers, like ordinary numbers, can be used in arithmetic operations (for example, addition and multiplication) that give another fuzzy number as the result. It should be noted that fuzzy uncertainty is optional and the other two options are nonfuzzy uncertainty expressed as ordinary certainty factors and absolute certainty ($CF = 1$). The methods of handling fuzzy uncertainty and fuzzy numbers are explained elsewhere in this article.

Rules management module. A rule is defined as an implication statement expressing the relationship between a set of antecedent propositions and a set of consequent propositions. Attached to each rule is a fuzzy/nonfuzzy uncertainty describing the degree of confidence in the rule. The antecedent part of a rule consists of a single proposition or any combination of two or more propositions connected by either a logical AND or a logical OR. But the consequent part of a rule can contain only a single proposition or multiple propositions with AND conjunctions between them. The reason for this is that the applicability of a rule with OR conjunctions in its consequent part is limited when backward reasoning (chaining) is employed. The following is an example of a rule with multiple propositions:

```
rule01: IF (the body is well-built
           OR the height is tall)
           AND
           the person is healthy
      THEN the weight of the person
           is heavy
      WITH CERTAINTY → close
           to 1.0
```

System properties management module. This module is responsible for the manipulation of the system properties of a knowledge base in System Z-II. The following are some system properties:

- **GOAL-OBJECTS:** specifies goal objects for the knowledge base

- **INITIAL-ASK-OBJECTS:** specifies objects whose values are to be asked at the start of a consultation
- **DOMAIN-DESCRIPTION:** stores the descriptions for the domain

Consultation driver

Inference engine. The inference engine of Z-II uses backward chaining to build the appropriate reasoning trees, but uses forward evaluation of the values of the fuzzy

terms. It can handle rules with multiple propositions, and it uses evidence combination for cases in which two or more rules have the same consequent proposition.

Reasoning chain. System Z-II adopts backward reasoning during the consulta-

Fuzzy numbers

A fuzzy number is a real-number fuzzy set that is both convex and normal. The following is the definition of a convex fuzzy set F :

$$\forall x, y \in R: \mu_F[\lambda x + (1 - \lambda)y] \geq \mu_F(x) \wedge \mu_F(y) \quad \forall \lambda \in [0, 1]$$

where R is the set of real numbers, and x, y , and μ are real numbers. A fuzzy set is normal if and only if the highest value of the degree of membership equals 1.0.

Expert systems can use fuzzy numbers to handle fuzziness or imprecision in real numbers and thus to represent and manipulate linguistic terms such as *near 0.6* and *close to 3*. In System Z-II fuzzy numbers represent the fuzzy uncertainty associated with a rule or a fact.

Fuzzy numbers, like ordinary numbers, can be used in arithmetic operations (addition, multiplication) that give another fuzzy number as the result (see the figure below). The formulas of some fuzzy number arithmetic operators are as follows:

fuzzy number addition +:

$$\mu_{A+B}(z) = V_{z=x+y}(\mu_A(x) \wedge \mu_B(y))$$

fuzzy number subtraction -:

$$\mu_{A-B}(z) = V_{z=x-y}(\mu_A(x) \wedge \mu_B(y))$$

fuzzy number multiplication *:

$$\mu_{A*B}(z) = V_{z=x*y}(\mu_A(x) \wedge \mu_B(y))$$

fuzzy number division /:

$$\mu_{A/B}(z) = V_{z=x/y}(\mu_A(x) \wedge \mu_B(y))$$

minimum of fuzzy numbers min_fn:

$$\mu_{\min\_fn(A,B)}(z) = V_{z=x \wedge y}(\mu_A(x) \wedge \mu_B(y))$$

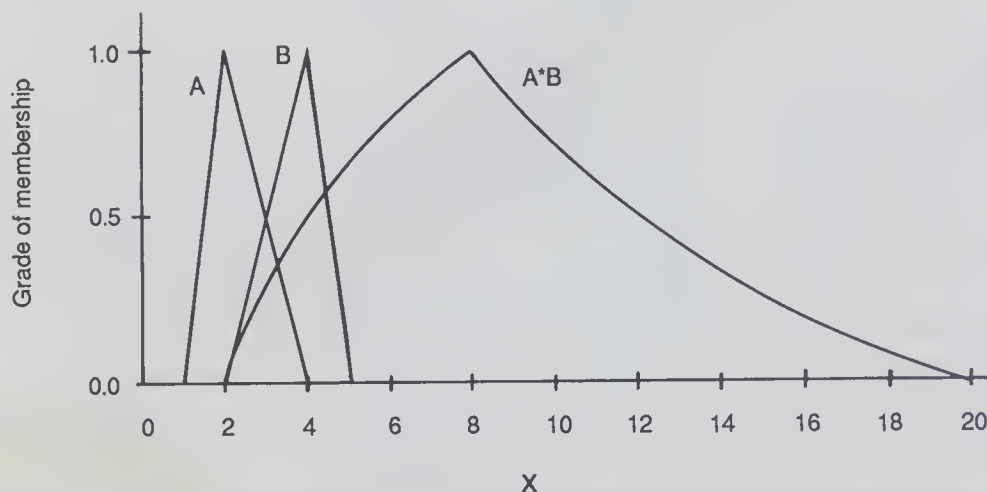
maximum of fuzzy numbers max_fn:

$$\mu_{\max\_fn(A,B)}(z) = V_{z=x \vee y}(\mu_A(x) \vee \mu_B(y))$$

where

A and B : fuzzy numbers
 μ : membership distribution function
 x, y , and z : real numbers
 $\vee$ and $\vee$: taking the maximum
 $\wedge$: taking the minimum

In Z-II, fuzzy numbers are assumed generally to be trapezoidal, and they are implemented as a list of four numbers. It has been found that trapezoidal fuzzy numbers are adequate to capture the fuzzy uncertainties in human intuition. The above fuzzy arithmetic operations are implemented so that they can handle this approximate representation of fuzzy numbers.



Multiplication of two fuzzy numbers.

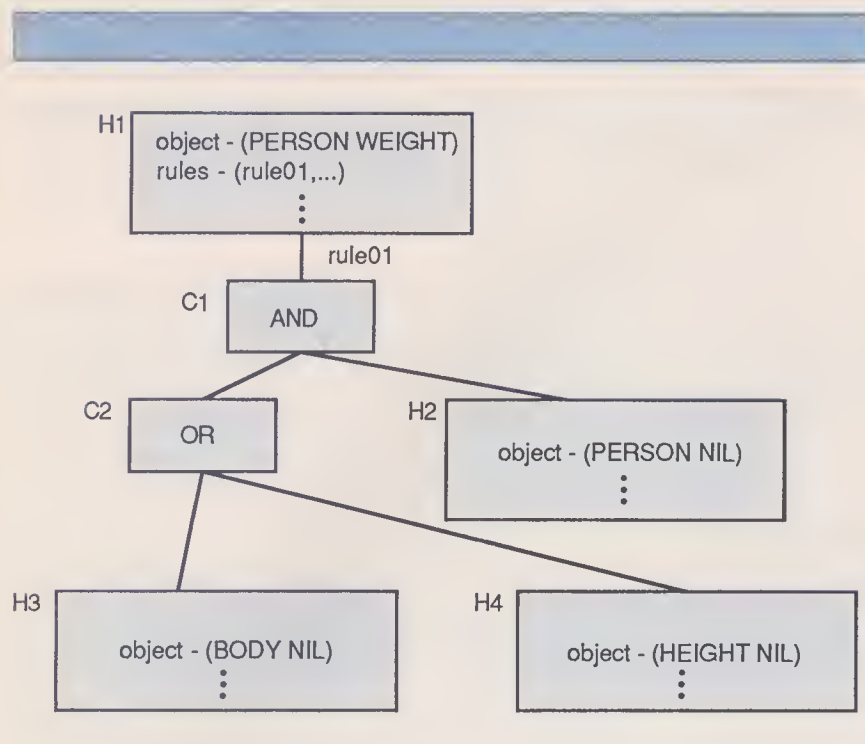


Figure 3. A history tree.

tion process because the questioning is guaranteed to follow the focused goal conclusion. The user may issue queries of the form

What should <OBJECT>
be?

e.g., What should the weight of the
person be?

The object in the query then becomes the current top-level goal. However, if the user selects an automatic mode of consultation, the system retrieves the objects stored in the GOAL-OBJECTS property of the current knowledge base and then considers each of these objects as the top-level goal. The system begins the reasoning by searching those rules whose consequent propositions have the goal object. This information can be retrieved from the content of the UPDATED-BY-RULES slot of the goal object. Each triggered rule is considered and a history tree is built at the same time.

Suppose there is a rule like the following:

rule01: IF (the body is well-built
OR the height is tall)
AND
the person is healthy
THEN the weight of the person
is heavy
WITH CERTAINTY $\rightarrow$ close
to 1.0

If the goal object is *the weight of the person*, the system triggers this rule and starts to examine its antecedent propositions. The resulting history tree is shown in Figure 3.

Based on the history tree, the system examines one of its antecedent objects, BODY NIL, at node H3. If the VALUE-LIST slot of this object is empty, the value of the object is not yet known and it becomes a subgoal object. Therefore, the system tries to obtain the value of the subgoal object by asking a question or by deducing from other rules and facts. The choice depends on the flag stored in the ASK-FIRST-OR-NOT slot of the subgoal

object. If the system decides to deduce the value from other rules and facts, the history tree continues to branch downward at node H3. Rules that have existed in ancestor nodes are ignored to prevent infinite looping in building the history tree.

After obtaining the value of the object at the subgoal node H3, the system chooses another subgoal node (H2 or H4) for consideration. The choice depends on two factors. One is the conjunctions between subgoal nodes. The other is whether the value of the object at node H3 is successfully matched by that of the available facts. Finally, the rule is fired if the antecedent part is satisfied completely. Thus, the system can find the value of the goal object by evaluating the rule and the matched fact.

If the objects involved are fuzzy, the inference engine uses fuzzy logic operations, calculating the fuzzy uncertainty of the goal object from the fuzzy uncertainties of the facts and the rules.

Rule evaluation. Suppose there are a rule and a fact:

rule: IF A is V_1 THEN C is V_2 (FN_1)

fact: A is V_1' (FN_2)

conclusion: C is V_2' (FN_3)

A : antecedent object
 C : consequent object
 FN_1 : fuzzy number denoting
uncertainty of the rule
 FN_2 : fuzzy number denoting
uncertainty of the fact
 FN_3 : fuzzy number denoting
uncertainty of the
conclusion

$V_1, V_2,$
 V_1', V_2' : values

If the object A in the antecedent is non-fuzzy, V_1 and V_1' must be the same atomic symbol in order to apply this rule. Therefore, V_2' in the conclusion equals V_2 , and the fuzzy uncertainty FN_3 of the conclusion is calculated by means of the fuzzy number multiplication of FN_1 and FN_2 :

$$FN_3 = FN_1 * FN_2$$

where $*$ denotes a fuzzy number multiplication.⁵ The formula for fuzzy number multiplication is given in the sidebar on fuzzy numbers.

If both A and C are fuzzy objects, V_1 and V_2 are represented by fuzzy sets F_1 and F_2 respectively. We can form a fuzzy

relation R by performing some fuzzy operations on F_1 and F_2 . The default method adopted in System Z-II for forming the fuzzy relation R is the R_{SG} approach proposed by Mizumoto, Fukami, and Tanaka,⁸ which has been found to be closer to human intuition and reasoning than other methods. However, the fuzzy relation can be selected from other options available in Z-II, such as R_S and R_G , or it can even be supplied by the user. The following are the three approaches to the fuzzy relation available in Z-II:

$$\begin{aligned} R_S &= F_1 \times V \rightarrow_S U \times F_2 \\ R_G &= F_1 \times V \rightarrow_G U \times F_2 \\ R_{SG} &= (F_1 \times V \rightarrow_S U \times F_2) \wedge (\sim F_1 \times V \rightarrow_G U \times \sim F_2) \end{aligned}$$

where

$$\mu_{F_1}(u) \rightarrow_S \mu_{F_2}(v) = \begin{cases} 1 & \text{if } \mu_{F_1}(u) \leq \mu_{F_2}(v) \\ 0 & \text{if } \mu_{F_1}(u) > \mu_{F_2}(v) \end{cases}$$

$$\mu_{F_1}(u) \rightarrow_G \mu_{F_2}(v) = \begin{cases} 1 & \text{if } \mu_{F_1}(u) \leq \mu_{F_2}(v) \\ \mu_{F_2}(v) & \text{if } \mu_{F_1}(u) > \mu_{F_2}(v) \end{cases}$$

U and V : the universe of discourse of F_1 and F_2 respectively

u and v : the elements of a fuzzy set

$\wedge$: intersection of two fuzzy relations

$\sim$: complement of a fuzzy set

μ : membership function

$\times$: Cartesian product of two fuzzy sets

V_1' in the fact should also be a fuzzy value represented by a fuzzy set F_1' . The fuzzy set F_2' of V_2' in the conclusion is obtained by applying a fuzzy composition operation (denoted by $\circ$) on F_1' and R : $F_2' = F_1' \circ R$. The calculation of the fuzzy uncertainty FN_3 of the conclusion is the same.

If A is fuzzy and C is nonfuzzy, V_2' in the conclusion must equal V_2 . However, the fuzzy uncertainty FN_3 is obtained by fuzzy multiplication of FN_1 , FN_2 , and the similarity M between F_1 and F_1' , which are the fuzzy sets of V_1 and V_1' respectively:

$$FN_3 = (FN_1 * FN_2 * M)$$

The similarity M is calculated by the following algorithm:

IF $N(F_1|F_1') > 0.5$
THEN $M = P(F_1|F_1')$
ELSE $M = (N(F_1|F_1') + 0.5) * P(F_1|F_1')$

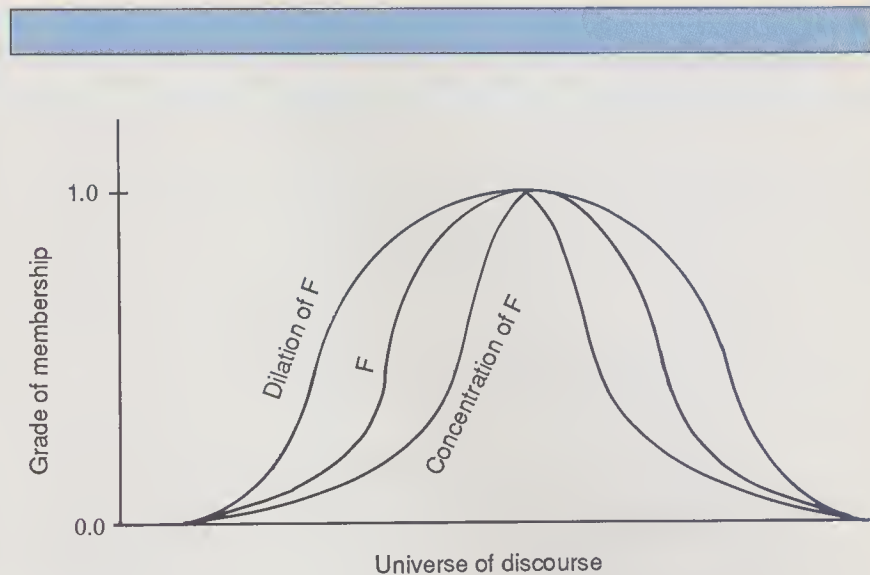


Figure 4. Dilation and concentration of F .

where $P(F_1|F_1')$ is the possibility of the fuzzy data F_1' given the fuzzy pattern F_1 , and $N(F_1|F_1')$ is the necessity of the fuzzy data F_1' given the fuzzy pattern F_1 .

The following are the formulas of the possibility and necessity measures between the fuzzy data and the fuzzy pattern:

$$\begin{aligned} \text{possibility:} \\ P(F_1|F_1') &= \max(\min(\mu_{F_1}(w), \mu_{F_1'}(w))) \end{aligned}$$

$$\begin{aligned} \text{necessity:} \\ N(F_1|F_1') &= 1 - P(\bar{F}_1|\bar{F}_1') \end{aligned}$$

where μ is the membership distribution function, w is the element in the universe of discourse of the above fuzzy sets, and $\bar{F}_1$ is the complement of F_1 .

The possibility between the fuzzy pattern F_1 and the fuzzy data F_1' gives the maximum of their intersection and measures to what extent they overlap. Under normal circumstances the necessity reflects the following relationships between two fuzzy sets:

$$\begin{aligned} N(F_1|F_1') > 0.5 &\Leftrightarrow F_1' \text{ is a concentration of } F_1 \\ N(F_1|F_1') = 0.5 &\Leftrightarrow F_1' \text{ is a duplicate of } F_1 \\ N(F_1|F_1') < 0.5 &\Leftrightarrow F_1' \text{ is a dilation of } F_1 \end{aligned}$$

As shown in Figure 4, if F_1' is a concentration of F_1 , it means that F_1' has a more concentrated or narrower distribution than F_1 . If V_1 and V_1' are the fuzzy terms in the rule and the matched fact respectively, then F_1 and F_1' represent two similar fuzzy concepts such as *good* and *very good* respectively. However, the more concentrated distribution of F_1' represents a more strongly expressed or stressed term than that of F_1 . For a dilation the situation is exactly the opposite.

On the other hand, the similarity measures how similar two fuzzy concepts represented by the two fuzzy sets are. When $N(F_1|F_1')$ is larger than 0.5, the similarity becomes saturated and is forced to equal the possibility (usually equal to one for two similar concepts). Examples:

(1) The fuzzy data is a concentration of the fuzzy pattern (i.e., necessity > 0.5).

rule: IF X is tall,
THEN X should be chosen as a member of the basketball team.
($CF_1 = 1.0$)

fact: X is very tall.
($CF_2 = 1.0$)

conclusion: X is chosen as a member of the basketball team.
($CF_3 = y$)

A sample case

The sample expert system described here illustrates the main features of System Z-II. Designed to help students choose their university department, this expert system has about 60 rules in its knowledge base. The following are some typical rules used in the system (the complete list appears later in this sidebar, under "Rules of the sample expert system"):

RULE CODE: phy-r5

IF your interest in reading science books or magazines is high

THEN your interest in investigating science should be high
WITH CERTAINTY → 0.9

RULE CODE: med-r1

IF your overall examination performance in medicine is good

THEN my recommendation for your choice of a university department should be medicine
WITH CERTAINTY → 0.9

RULE CODE: test-r1

IF the mark you obtained in the English Usage Test is <50

THEN your English result should be bad
WITH CERTAINTY → 1.0

RULE CODE: exam-r3

IF your physics result is very good AND
your mathematics result is good

THEN your overall examination performance in physics should be good
WITH CERTAINTY → 1.0

RULE CODE: soc-r4

IF your interest in analyzing human behavior is high

THEN your overall interest in social science should be high
WITH CERTAINTY → roughly 0.85

Rule phy-r5 has the fuzzy concept *high* in its antecedent and consequence. Rule med-r1 has the fuzzy concept *good* in its antecedent and a nonfuzzy consequence. Rule test-r1 has a fuzzy consequence and a nonfuzzy antecedent, which is also a numeric-comparison logic control. Rule exam-r3 demonstrates multiple propositions in its antecedent. Some of the rules are not absolutely certain. Thus, a certainty factor, which can be fuzzy or nonfuzzy, is attached to each rule.

During a consultation, the system requests the user to enter the appropriate fuzzy or nonfuzzy facts. After obtaining enough facts, the system presents its conclusions. For a multivalued goal, the conclusions are given in descending order according to their certainties. The following is a sample consultation, including facts and conclusions:

** The following are the facts you entered:

1. The fact that you have a risk-taking personality is false. (1.0)
2. Your interest in the business environment is very low. (about 0.9)
3. Your Chinese result is fair. (1.0)
4. The mark you obtained in the English Usage Test is 76.0. (1.0)
5. Your mathematics result is good. (1.0)
6. The preferred career as a teacher is true. (1.0)
7. The preferred career as a researcher is true. (1.0)
8. Your interest in analyzing human behavior is medium. (around 0.8)

9. Your interest in reading fiction is more or less low. (1.0)
10. The preferred career as a doctor is false. (1.0)
11. Your interest in analyzing the human body is low. (0.7)
12. Your chemistry result is fair. (1.0)
13. Your biology result is fair. (1.0)
14. Your physics result is very good. (1.0)
15. Your interest in manipulating mathematical symbols is rather high. (1.0)
16. The preferred career as an electronics engineer is false. (1.0)
17. Your interest in building electronic kits is medium. (1.0)
18. The preferred career as a programmer is true. (1.0)
19. Your interest in writing computer programs is high. (1.0)
20. Your interest in computers is high. (1.0)
21. Your interest in reading science books or magazines is very high. (1.0)
22. Your interest in performing chemical experiments is medium. (1.0)
23. Your interest in analyzing chemical substances is more or less high. (0.9)
24. Your interest in observing animals is high. (1.0)
25. Your interest in observing plants is high. (1.0)

** After analyzing your responses, Z-II makes the following conclusions in preference order for your choice of a university department:

It is 0.97 certain that — the department of physics
It is 0.95 certain that — the department of computer science
It is 0.93 certain that — the department of mathematics
It is 0.90 certain that — the department of electronics
It is 0.86 certain that — the department of biology
It is 0.72 certain that — the department of chemistry
It is very close to 0.70 certain that — the faculty of arts
It is very close to 0.69 certain that — the faculty of social science
It is close to 0.49 certain that — the faculty of business
It is 0.48 certain that — the faculty of medicine

** It is 0.9 certain that the chance of your entering the physics department is fuzzily very high.

In this consultation, two goals can be changed easily in the expert system. The first one is *department selection*, which is a multivalued nonfuzzy goal; the other is *the chance of entering the physics department*, a single-valued fuzzy goal. The first goal mainly depends on factors such as the student's examination performance, the student's suitability for a future job in the field, and the student's overall interests. However, the second goal depends only on the performance in the physics examination, and its conclusion is a fuzzy term *fuzzily very high*. The membership distribution of the fuzzy set of this term is similar to that of *very high* except that it has a much gentler slope.

Rules of the sample expert system

RULE CODE: exam-r1

IF your biology result is very good AND
your chemistry result is good AND
(your mathematics result is fair or good OR
your physics result is fair or good)

THEN your overall examination performance in biology should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r2

IF your chemistry result is very good AND
(your mathematics result is fair or good OR
your physics result is fair or good)
THEN your overall examination performance in chemistry
should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r3

IF your physics result is very good AND
your mathematics result is good
THEN your overall examination performance in physics
should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r4

IF your mathematics result is very good AND
your physics result is good
THEN your overall examination performance in computer
science should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r5

IF your mathematics result is good AND
your physics result is good
THEN your overall examination performance in electronics
should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r6

IF your mathematics result is very good AND
your physics result is fair or good
THEN your overall examination performance in mathematics
should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r7

IF your chemistry result is very good AND
your biology result is very good AND
your mathematics result is good AND
your physics result is good
THEN your overall examination performance in medicine
should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r8

IF your Chinese result is good AND
your English result is good
THEN your overall examination performance in arts should be
good
WITH CERTAINTY → 1.0

RULE CODE: exam-r9

IF your Chinese result is good AND
your English result is good AND
your mathematics result is fair or good
THEN your overall examination performance in social science
should be good
WITH CERTAINTY → 1.0

RULE CODE: exam-r10

IF your Chinese result is good AND
your English result is good AND
your mathematics result is good
THEN your overall examination performance in business
should be good
WITH CERTAINTY → 1.0

RULE CODE: phy-r1

IF your overall examination performance in physics is
good
THEN your chance of entering physics should be high AND
my recommendation for your choice of a university
department should be physics
WITH CERTAINTY → 0.9

RULE CODE: phy-r2

IF your overall interest in physics is high
THEN my recommendation for your choice of a university
department should be physics
WITH CERTAINTY → 0.6

RULE CODE: phy-r3

IF the overall suitability of a future job in physics is good
THEN my recommendation for your choice of a university
department should be physics
WITH CERTAINTY → 0.4

RULE CODE: phy-r4

IF your interest in investigating science is high
THEN your overall interest in physics should be high
WITH CERTAINTY → 1.0

RULE CODE: phy-r5

IF your interest in reading science books or magazines is
high
THEN your interest in investigating science should be high
WITH CERTAINTY → 0.9

RULE CODE: mth-r1

IF your overall examination performance in mathematics
is good
THEN my recommendation for your choice of a university
department should be mathematics
WITH CERTAINTY → 0.9

RULE CODE: mth-r2

IF your overall interest in mathematics is high
THEN my recommendation for your choice of a university
department should be mathematics
WITH CERTAINTY → 0.6

RULE CODE: mth-r3

IF the overall suitability of a future job in mathematics is
good
THEN my recommendation for your choice of a university
department should be mathematics
WITH CERTAINTY → 0.4

RULE CODE: mth-r4

IF your interest in manipulating mathematical symbols is
high
THEN your overall interest in mathematics should be high
WITH CERTAINTY → 0.85

RULE CODE: csc-r1

IF your overall examination performance in computer
science is good
THEN my recommendation for your choice of a university
department should be computer science
WITH CERTAINTY → 0.9

RULE CODE: csc-r2

IF your overall interest in computer science is high
THEN my recommendation for your choice of a university
department should be computer science
WITH CERTAINTY → 0.6

RULE CODE: csc-r3
 IF the overall suitability of a future job in computer science is good
 THEN my recommendation for your choice of a university department should be computer science
 WITH CERTAINTY → 0.4

RULE CODE: csc-r4
 IF your interest in computers is high
 THEN your overall interest in computer science should be high
 WITH CERTAINTY → 0.8

RULE CODE: csc-r5
 IF your interest in writing computer programs is high
 THEN your overall interest in computer science should be high
 WITH CERTAINTY → 0.8

RULE CODE: bio-r1
 IF your overall examination performance in biology is good
 THEN my recommendation for your choice of a university department should be biology
 WITH CERTAINTY → 0.9

RULE CODE: bio-r2
 IF your overall interest in biology is high
 THEN my recommendation for your choice of a university department should be biology
 WITH CERTAINTY → 0.6

RULE CODE: bio-r3
 IF the overall suitability of a future job in biology is good
 THEN my recommendation for your choice of a university department should be biology
 WITH CERTAINTY → 0.4

RULE CODE: bio-r4
 IF your interest in investigating living things is high
 THEN your overall interest in biology should be high
 WITH CERTAINTY → 1.0

RULE CODE: bio-r5
 IF your interest in observing plants is high
 THEN your interest in investigating living things should be high
 WITH CERTAINTY → 0.8

RULE CODE: bio-r6
 IF your interest in observing animals is high
 THEN your interest in investigating living things should be high
 WITH CERTAINTY → 0.8

RULE CODE: chm-r1
 IF your overall examination performance in chemistry is good
 THEN my recommendation for your choice of a university department should be chemistry
 WITH CERTAINTY → 0.9

RULE CODE: chm-r2
 IF your overall interest in chemistry is high
 THEN my recommendation for your choice of a university department should be chemistry
 WITH CERTAINTY → 0.6

RULE CODE: chm-r3
 IF the overall suitability of a future job in chemistry is good
 THEN my recommendation for your choice of a university

department should be chemistry
 WITH CERTAINTY → 0.4

RULE CODE: chm-r4
 IF your interest in analyzing chemical substances is high
 THEN your overall interest in chemistry should be high
 WITH CERTAINTY → 0.8

RULE CODE: chm-r5
 IF your interest in performing chemical experiments is high
 THEN your overall interest in chemistry should be high
 WITH CERTAINTY → 0.8

RULE CODE: ele-r1
 IF your overall examination performance in electronics is good
 THEN my recommendation for your choice of a university department should be electronics
 WITH CERTAINTY → 0.9

RULE CODE: ele-r2
 IF your overall interest in electronics is high
 THEN my recommendation for your choice of a university department should be electronics
 WITH CERTAINTY → 0.6

RULE CODE: ele-r3
 IF the overall suitability of a future job in electronics is good
 THEN my recommendation for your choice of a university department should be electronics
 WITH CERTAINTY → 0.4

RULE CODE: ele-r4
 IF your interest in building electronic kits is high
 THEN your overall interest in electronics should be high
 WITH CERTAINTY → 0.9

RULE CODE: art-r1
 IF your overall examination performance in arts is good
 THEN my recommendation for your choice of a university department should be arts
 WITH CERTAINTY → 0.9

RULE CODE: art-r2
 IF your overall interest in arts is high
 THEN my recommendation for your choice of a university department should be arts
 WITH CERTAINTY → 0.6

RULE CODE: art-r3
 IF the overall suitability of a future job in arts is good
 THEN my recommendation for your choice of a university department should be arts
 WITH CERTAINTY → 0.4

RULE CODE: art-r4
 IF your interest in reading fiction is high
 THEN your overall interest in arts should be high
 WITH CERTAINTY → around 0.9

RULE CODE: ba-r1
 IF your overall examination performance in business is good
 THEN my recommendation for your choice of a university department should be business
 WITH CERTAINTY → 0.9

RULE CODE: ba-r2
 IF your overall interest in business is high
 THEN my recommendation for your choice of a university department should be business

WITH CERTAINTY → 0.6

RULE CODE: ba-r3

IF the overall suitability of a future job in business is good
THEN my recommendation for your choice of a university department should be business
WITH CERTAINTY → 0.4

RULE CODE: ba-r4

IF your interest in the business environment is high
THEN your overall interest in business should be high
WITH CERTAINTY → 0.9

RULE CODE: med-r1

IF your overall examination performance in medicine is good
THEN my recommendation for your choice of a university department should be medicine
WITH CERTAINTY → 0.9

RULE CODE: med-r2

IF your overall interest in medicine is high
THEN my recommendation for your choice of a university department should be medicine
WITH CERTAINTY → 0.6

RULE CODE: med-r3

IF the overall suitability of a future job in medicine is good
THEN my recommendation for your choice of a university department should be medicine
WITH CERTAINTY → 0.4

RULE CODE: med-r4

IF your interest in analyzing the human body is high
THEN your overall interest in medicine should be high
WITH CERTAINTY → 0.95

RULE CODE: soc-r1

IF your overall examination performance in social science is good
THEN my recommendation for your choice of a university department should be social science
WITH CERTAINTY → 0.9

RULE CODE: soc-r2

IF your overall interest in social science is high
THEN my recommendation for your choice of a university department should be social science
WITH CERTAINTY → 0.6

RULE CODE: soc-r3

IF the overall suitability of a future job in social science is good
THEN my recommendation for your choice of a university department should be social science
WITH CERTAINTY → 0.4

RULE CODE: soc-r4

IF your interest in analyzing human behavior is high
THEN your overall interest in social science should be high
WITH CERTAINTY → roughly 0.85

RULE CODE: job-r1

IF the preferred career as a teacher is desirable OR the preferred career as a researcher is desirable
THEN the overall suitability of a future job in biology should be good AND the overall suitability of a future job in chemistry should be good AND

the overall suitability of a future job in physics should be good AND the overall suitability of a future job in mathematics should be good AND the overall suitability of a future job in arts should be good AND the overall suitability of a future job in social science should be good

WITH CERTAINTY → 1.0

RULE CODE: job-r2

IF the preferred career as a programmer is desirable
THEN the overall suitability of a future job in computer science should be good
WITH CERTAINTY → 1.0

RULE CODE: job-r3

IF the preferred career as an electronics engineer is desirable
THEN the overall suitability of a future job in electronics should be good
WITH CERTAINTY → 1.0

RULE CODE: job-r4

IF the preferred career as a doctor is desirable
THEN the overall suitability of a future job in medicine should be good
WITH CERTAINTY → 1.0

RULE CODE: job-r5

IF the fact that you have a risk-taking personality is true
THEN the overall suitability of a future job in business should be good
WITH CERTAINTY → 0.95

RULE CODE: test-r1

IF the mark you obtained in the English Usage Test is < 50.0
THEN your English result should be bad
WITH CERTAINTY → 1.0

RULE CODE: test-r2

IF the mark you obtained in the English Usage Test is < 70.0 AND the mark you obtained in the English Usage Test is ≥ 50.0
THEN your English result should be fair
WITH CERTAINTY → 1.0

RULE CODE: test-r3

IF the mark you obtained in the English Usage Test is < 80.0 AND the mark you obtained in the English Usage Test is ≥ 70.0
THEN your English result should be quite good
WITH CERTAINTY → 1.0

RULE CODE: test-r4

IF the mark you obtained in the English Usage Test is < 90.0 AND the mark you obtained in the English Usage Test is ≥ 80.0
THEN your English result should be good
WITH CERTAINTY → 1.0

RULE CODE: test-r5

IF the mark you obtained in the English Usage Test is ≥ 90.0
THEN your English result should be very good
WITH CERTAINTY → 1.0

Therefore, F_1 is the fuzzy set of *tall*, while F_1' is the fuzzy set of *very tall*. If the membership distributions of F_1 and F_1' equal that in Figure 1, then

$$\begin{aligned} N(F_1|F_1') &= 0.6 & (> 0.5) \\ P(F_1|F_1') &= 1.0 \end{aligned}$$

Based on the above algorithm,

$$M = P(F_1|F_1') = 1.0$$

As a result, the certainty factor of the conclusion,

$$\begin{aligned} y &= CF_1 * CF_2 * M \\ &= 1.0 * 1.0 * 1.0 = 1.0 \end{aligned}$$

(2) If the fuzzy data is a dilation of the fuzzy pattern (i.e., necessity < 0.5), then the similarity should depend on the necessity and possibility measures.

Assume the fact in the above example is changed to

fact: X is quite tall. ($CF_2 = 1.0$)

Now, F_1' is the fuzzy set representing *quite tall* and if its membership distribution equals that in Figure 1, then

$$\begin{aligned} N(F_1|F_1') &= 0.3 & (< 0.5) \\ P(F_1|F_1') &= 1.0 \end{aligned}$$

Based on the above algorithm,

$$\begin{aligned} M &= (N(F_1|F_1') + 0.5) * P(F_1|F_1') \\ &= (0.3 + 0.5) * 1.0 = 0.8 \end{aligned}$$

As a result, the certainty factor of the conclusion,

$$y = 1.0 * 1.0 * 0.8 = 0.8$$

Rules with multiple propositions. In Z-II the consequent part of a rule can contain only multiple propositions ($C_1, C_2, \dots, C_n$) with AND conjunctions between them. They can be treated as multiple rules with a single conclusion. So the following rule:

IF antecedent-propositions,
THEN C_1 AND C_2 AND $\dots$ C_n

is equivalent to the following rules:

IF antecedent-propositions, THEN C_1
IF antecedent-propositions, THEN C_2

•
•
•

IF antecedent-propositions, THEN C_n

Therefore, only the problem of multiple propositions in the antecedent with a single proposition in the consequence needs to be considered. If the object in the consequent proposition is nonfuzzy, no special treatment is needed. However, if the consequent proposition is fuzzy, the fuzzy set of the value V_3' in the conclusion is

calculated with the following two basic algorithms¹⁰:

(1) rule: IF A_1 AND A_2 ,
THEN C is V_3
facts: A_1', A_2'
conclusion: C is V_3'
algorithm: The fuzzy set representing V_3' in the conclusion C is obtained by taking fuzzy union of the fuzzy sets of F_1 and F_2

where the fuzzy set F_1 is obtained from the composition operation on the single rule (IF A_1 , THEN C is V_3) and the fact A_1' , while the fuzzy set F_2 is obtained from the composition operation on the single rule (IF A_2 , THEN C is V_3) and the fact A_2' . Union operation is used on F_1 and F_2 because they have an OR relation between them after we break up the rule [$(A_1 \text{ AND } A_2) \rightarrow C$] into [$(A_1 \rightarrow C) \text{ OR } (A_2 \rightarrow C)$] by the distribution law in classic logic.

(2) rule: IF A_1 OR A_2 ,
THEN C is V_3
facts: A_1', A_2'
conclusion: C is V_3'
algorithm: The same as above except that fuzzy intersection rather than union should be applied on the fuzzy sets of F_1 and F_2

where

A_1, A_2 : antecedent propositions that can be single or multiple
 C : object in the consequent proposition
 A_1', A_2' : data propositions (facts)
 V_3, V_3' : values

fuzzy intersection: IF D is the fuzzy intersection of F_1 and F_2 , THEN
 $\mu_D(x) = \min(\mu_{F_1}(x), \mu_{F_2}(x))$

fuzzy union: IF D is the fuzzy union of F_1 and F_2 , THEN
 $\mu_D(x) = \max(\mu_{F_1}(x), \mu_{F_2}(x))$

D, F_1, F_2 : fuzzy sets

The above two algorithms can be applied repeatedly to handle any combination of antecedent propositions. For instance:

rule: IF (the raw material cost is low OR the production cost is low) AND sales are high
THEN the profit should be good

facts: The raw material cost is very low
The production cost is low
Sales are rather high

where *low*, *high*, *good*, *very low*, and *rather high* are fuzzy concepts.

Let F_1 be the fuzzy set obtained by making an inference from the single rule

IF the raw material cost is low THEN the profit should be good
and the fact
The raw material cost is very low ;

F_2 be the fuzzy set obtained by making an inference from the single rule

IF the production cost is low THEN the profit should be good
and the fact
The production cost is low ; and

F_3 be the fuzzy set obtained by making an inference from the single rule

IF sales are high
THEN the profit should be good
and the fact
Sales are rather high

The fuzzy set F representing the fuzzy value of the object *profit* in the conclusion is determined as follows:

F = fuzzy union between F_{12} and F_3 ,
where
 F_{12} = fuzzy intersection between F_1 and F_2

As a result, F will indicate the fuzzy concept *good* and the conclusion *The profit is good* is drawn.

The fuzzy uncertainty of the conclusion deduced from rules with multiple-

antecedent propositions is calculated by means of fuzzy number arithmetic operators¹ in formulas used by Mycin's CF model. For example:

rule: IF A_1 and A_2 then C (FN_R)
 facts: A_1' (FN_1)
 A_2' (FN_2)

conclusion: C' (FN_C)

$$FN_C = (\min\_fn(FN_1, FN_2)) * FN_R$$

A_1, A_2 : antecedent propositions that can be single or multiple

C : consequent proposition

C' : conclusion

FN_R : fuzzy uncertainty of the rule

FN_1, FN_2 : fuzzy uncertainties of the facts

FN_C : fuzzy uncertainty of the conclusion

$\min\_fn$: take the minimum of two fuzzy numbers

$*$: fuzzy number multiplication

A_1', A_2' : data propositions (facts)

If logical OR is used, the calculation is the same except that the fuzzy maximum is taken rather than the minimum. For any combination of antecedent propositions, the two calculations can be applied repeatedly to handle fuzzy uncertainties corresponding to the matched facts and the rule. (For fuzzy operations, see the sidebar on fuzzy numbers.)

Evidence combination. In some cases, two or more rules have the same consequent proposition. Each of these rules with matched facts can be treated as contributing evidence toward the conclusion. A conclusion C_R can be drawn from the evidence contributed by these rules and facts. For instance:

rules: r_1 —if A_1 then C
 r_2 —if A_2 then C
 facts: $-A_1', A_2'$

conclusion: C_R , obtained from
 $-C'$ (FN_1)
 and $-C''$ (FN_2)

where

r_1, r_2 : rule codes

A_1, A_2 : antecedent propositions

C : consequent proposition

C', C'' : conclusions from r_1 & A_1' and r_2 & A_2' respectively

FN_1, FN_2 : fuzzy uncertainties of the conclusion

If the object involved in the consequent proposition is fuzzy, the fuzzy sets corresponding to conclusions C' and C'' , obtained from performing approximate reasoning on the respective rules and facts, are combined by taking the fuzzy intersection between them to obtain the fuzzy set corresponding to the combined conclusion C_R . The operation can be applied repeatedly if there are more than two rules with respective matching facts but the same consequent proposition.

The fuzzy uncertainties of the respective conclusions C' and C'' are also aggregated to form an overall uncertainty FN_R for C_R . Basically, two uncertainties are considered at each time and combined according to the following formula, similar to the evidence combination formula in Mycin's CF model:

$$FN_R = FN_1 + FN_2 - (FN_1 * FN_2)$$

where

FN_R : the combined fuzzy uncertainty

FN_1, FN_2 : fuzzy uncertainties of C' and C'' respectively

$+$: fuzzy number addition

$-$: fuzzy number subtraction

$*$: fuzzy number multiplication

If there are more than two rules with the same consequent proposition, this formula is repeatedly applied until an overall fuzzy certainty is obtained.

Linguistic approximation routine. Linguistic approximation is a process that maps the set of fuzzy sets onto a set of linguistic values or expressions. In Z-II this process is needed for two purposes. One is to find the corresponding verbal descriptions of fuzzy sets representing fuzzy values. The other is to get the linguistic descriptions of fuzzy numbers representing fuzzy uncertainties, which is an original idea presented in this article.

The technique adopted in the linguistic approximation makes use of two factors: the imprecision and the location of a fuzzy set. The *imprecision* of a fuzzy set is defined as the sum of membership values; the *location* is the center of gravity. The possibility distribution of each linguistic value can be uniquely identified by the imprecision and location of a fuzzy set; the corresponding linguistic value can be matched and selected accordingly.

Review management module. A user can review the case data (facts) at any time during the consultation process and can modify case data after a review. The review management module provides routines to monitor and trace the relevant rules and facts.

This module is also responsible for tracing the reasoning chain when explanations are required. The system provides two types of explanations: *why* a fact is required by the system and *how* a fact is established.

Finally, this module can handle *what-if* reviews, which find out what conclusions will be deduced if certain facts are changed.

Fuzzy knowledge base

The fuzzy knowledge base is responsible for storing the knowledge entities, such as the objects, rules, and fuzzy terms of a knowledge base, acquired through the knowledge acquisition subsystem. These knowledge entities, representing expertise, provide information that enables the inference engine to perform consultations.

The fuzzy knowledge base is implemented with hash tables and property lists. These two data structures provided by Common Lisp are suitable for implementing expert systems in general. In System Z-II several hash tables are used to store objects, fuzzy terms, and rules. Accessing data from a hash table is efficient because the built-in hashing function is transparent to users.⁹ Property lists store the slots containing various types of knowledge entities.

The most distinctive feature of the fuzzy knowledge base is that it also stores fuzzy sets representing fuzzy terms. Each fuzzy set is implemented as a list of numbers that represent grades of membership (possibility values) on an imaginary psychological continuum with an interval scale.

To facilitate better knowledge engineering and simulation of human reasoning, we have built fuzzy concepts and inexact reasoning into System Z-II, an expert system shell based on fuzzy logic and fuzzy numbers. The added features of fuzzy concepts and inexact reasoning are particularly indispensable for building expert systems in application areas such as risk analysis and psychoanalysis because imprecision and

fuzziness are always present in the natural-language expressions of domain experts and end users in these fields. The system also exploits the power of fuzzy logic in the natural-language user interface. Z-II can handle both fuzziness and uncertainty, the two basic inexact concepts. Fuzzy sets and relations deal with the fuzziness in approximate reasoning, while fuzzy numbers manipulate the uncertainty. The system gains power by allowing any mix of fuzzy and normal terms, numeric-comparison logic controls, and uncertainties.

System Z-II has been used to construct several expert systems in university department selection, medical diagnosis, psychoanalysis, and risk analysis. These systems were built with the aid of many experts, who found it natural and convenient to express their knowledge by means of the fuzzy concepts supported by this tool. The satisfactory performance of these systems has demonstrated the feasibility and effectiveness of introducing fuzzy concepts into expert systems. □

References

1. A. Kaufmann and M.M. Gupta, *Introduction to Fuzzy Arithmetic: Theory and Applications*, Van Nostrand Reinhold, New York, 1985.
2. K.P. Adlassnig and G. Kolarz, "Representation and Semi-automatic Acquisition of Medical Knowledge in Cadiag-1 and Cadiag-2," *Computers and Biomedical Research*, Vol. 19, 1986, pp. 63-79.
3. T. Whalen, B. Schott, and F. Ganoe, "Fault Diagnosis in a Fuzzy Network," *Proc. 1982 Int'l Conf. Cybernetics and Society*, IEEE Press, New York, 1982.
4. T.P. Martin, J.F. Baldwin, and B.W. Pilsworth, "The Implementation of FProlog—A Fuzzy Prolog Interpreter," *Fuzzy Sets and Systems*, Vol. 23, 1987, pp. 119-129.
5. B.G. Buchanan and E.H. Shortliffe, *Rule-Based Expert Systems*, Addison-Wesley, Reading, Mass., 1984.
6. J.L. O'Neill, "Plausible Reasoning," *Australian Computer J.*, Vol. 19, No. 1, 1987, pp. 2-15.
7. G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, N.J., 1976.
8. M. Mizumoto, S. Fukami, and K. Tanaka, "Some Methods of Fuzzy Reasoning," in *Advances in Fuzzy Set Theory and Applications*, M.M. Gupta, R.K. Ragade, and R.R. Yager, eds., North-Holland, Amsterdam, 1979, pp. 117-136.
9. K.S. Leung and W. Lam, "The Implementation of Fuzzy Knowledge-Based System Shells," *Proc. Tencon 87, IEEE Region 10 Conf.*, pp. 650-654.
10. T. Whalen and B. Schott, "Decision Support with Fuzzy Production Systems," in *Advances in Fuzzy Sets, Possibility Theory and Applications*, P.P. Wang, ed., Plenum Press, 1982.



K.S. Leung is a lecturer in the department of computer science at the Chinese University of Hong Kong. His research interests lie in the areas of expert systems and fuzzy logic applications in artificial intelligence.

Leung received his BS and PhD degrees from the University of London in 1977 and 1980, respectively. He is a member of IEEE, BCS, and IEE.



Raymond W. Lam is a research student in the department of computer science at the Chinese University of Hong Kong. His current research interests include expert systems and artificial intelligence.

Lam graduated from the Chinese University of Hong Kong with a BS degree in computer science.

1989 INTERNATIONAL SYMPOSIUM ON MULTIPLE-VALUED LOGIC CALL FOR PAPERS

PROGRAM COMMITTEE

AMERICAS

Dr. J. C. Muzio
Dept. of Computer Science
University of Victoria
P.O. Box 1700
Victoria, B.C.
Canada V8W 2Y2

EUROPE / AFRICA

Prof. C. Moraga
FB Informatik
Universitaet Dortmund
Postfach 500500
4600 Dortmund 50
West Germany

ASIA / PACIFIC

Prof. Mou Hu
Shanghai Institute of
Railway Technology
I, Zhen Nam Lu
Shanghai China, 201803

The Multiple-Valued Logic Technical Committee of the IEEE Computer Society will hold its 19th annual Symposium on May 29 - 31, 1989 in Guangzhou, China. The Symposium is sponsored by the IEEE Computer Society and the TC MVL of the Chinese Computer Federation and hosted by the South China University of Technology. You are invited to submit an original research, survey, or tutorial paper on any subject in the area of Multiple-Valued Logic. Authors are requested to submit five copies (in English) of their double-spaced typed manuscript on 8.5 by 11 inch or A4 paper by November 1, 1988. Each paper should include a 50 - 100 word abstract. Please submit full addresses, telephone numbers, email addresses, etc. for all authors. Papers should be sent to the closest Program Chair. Authors will be notified by February 1, 1989. Photo-ready copies of accepted papers are due by March 1, 1989.

For further information contact Dr. D. M. Miller, ISMVL-89 Co-Chairman, Department of Computer Science, University of Victoria, Canada V8W 2Y2 (604) 721-7220.



IEEE COMPUTER SOCIETY



TECHNICAL COMMITTEE ON
MULTIPLE-VALUED LOGIC

CALL FOR PAPERS

FTCS



The Nineteenth International
Symposium on Fault-Tolerant
Computing

"Fault Tolerance in Practice"

June 21-23, 1989

Hyatt Regency
Chicago, Illinois

Symposium Chairs

R. K. Iyer
J. A. Abraham
Univ. of Illinois, USA

Program Chair

S. M. Reddy
Univ. of Iowa, USA

Organizing Cmte. Chair

J. H. Patel
Univ. of Illinois, USA

Advisory Cmte. Chair

Y. Levendel
AT&T, USA

Publicity Chair

K. Saluja
Univ. of Wisconsin, USA

Publication Chair

W. K. Fuchs
Univ. of Illinois, USA

Registration Chair

P. Banerjee
Univ. of Illinois, USA

Local Arrangements Chair

S. Kania
AT&T, USA

Technical Program Committee

Y. Yang, China
Y. Tamir, USA
L. Stringini, Italy
T. B. Smith, USA
K. G. Shin, USA
J. Shen, USA
S. Seth, USA
R. Schlichting, USA
J. Rajski, Canada
D. Pradhan, USA
J. Patel, USA
T. Nanya, Japan
J. Meyer, USA
E. McCluskey, USA
B. Littlewood, UK
B. Krishnamurthy, USA
K. Kinoshita, Japan
K. Kim, USA
T. Kikuno, Japan
J. Kelly, USA
D. Johnson, USA
V. Iyengar, USA
B. Helvik, Norway
J. Hayes, USA
A. Goyal, USA
J. Goldberg, USA
W. K. Fuchs, USA
S. Elkind, USA
A. Dahbura, USA
M. Dal Cin, FRG
B. Courtois, France
E. Cerny, Canada
V. Basili, USA
J. Arlat, France

Ex Officio Member

D. A. Rennels
Univ. of California, USA

All papers should be sent to:

Prof. S. M. Reddy
Dept. of Electrical & Computer Engr.
Engineering Bldg., Capitol Street
University of Iowa
Iowa City, Iowa 52242, USA

FTCS is the international symposium on fault tolerant computing systems. It is devoted to state-of-the-art issues in fault tolerant computing and encompasses all aspects of specifying, designing, modeling, implementing, testing, diagnosing, and evaluating dependable and fault tolerant computing systems and their components. Papers of theoretical and practical importance are solicited. This year, the symposium will be held in Chicago, the heartland of the USA.

Major topics include but are not limited to:

[A] Fault-tolerant Systems: Networks, Processor Arrays, Real-time Systems, Spaceborne Systems, Multicomputers.

[B] Fault-tolerant Hardware: Memories, Processors, Self-checking Circuits, Repair and Reconfiguration Techniques.

[C] Fault-tolerant Software: Specifications and Languages, Recovery. Fault-tolerant Programs.

[D] Software Verification and Testing: Specification, Languages, Verification techniques, Testing and Maintenance.

[E] Modeling and Experimentation: Reliability, Availability and Performability Modeling, Experimental Analysis of Hardware and Software Dependability, Measurements on Real Systems, Fault injection.

[F] Testing: Fault Models, Test Generation, Coverage Analysis and Simulation, Test Methodologies, Testable Designs.

[G] Systems under Design: Fault-tolerance Goals, Methodologies, Project Status.

[II] Emerging Areas: CAD for Fault Tolerance, Neural Networks, Design for Reliability, Artificial Intelligence, Optical Computing.

Sponsored by:

- Technical committee on fault-tolerant computing, IEEE Computer Society
- University of Illinois at Urbana-Champaign
- IFIP WG 10.4

Information for Authors:

Six copies of the abstract of the paper, including up to five keywords, should be submitted before October 14, 1988. Submit six copies of the paper (double spaced) before the submission deadline, November 21, 1988. Papers should be no longer than 5000 words. The first page of each paper must include the title, the author's name, affiliations, complete mailing address, telephone number and electronic mail address where applicable, a maximum 150-word abstract of the paper and up to five keywords (important for correct classification of the paper). The first page should also indicate that the paper has been cleared through the author's affiliations. For multiple authors please indicate who will present the paper, if accepted. Submissions arriving late or departing significantly from these guidelines risk rejection without consideration of their merits.

Important Dates

October 14, 1988	November 21, 1988
<i>Abstract Due</i>	<i>Submission Deadline</i>
March 6, 1989	
<i>Acceptance Notification</i>	

In conjunction with the symposium, a workshop on Hardware Fault Tolerance in Multiprocessors on June 19-20 is planned at Urbana, Illinois. For more information, please contact P. Banerjee or W. K. Fuchs at the Univ. of Illinois at Urbana-Champaign.



THE COMPUTER SOCIETY
OF THE IEEE



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

National Conference on Strategic Management of R&D

Ware Myers, Contributing Editor

Marilyn Potes, Managing Editor

The nation badly needs leadership in laying out an overall national strategy for technology development," said Conference Chair Laurence J. Adams in his closing summary of the recent three-day National Conference on Strategic Management of Research and Development. "The system that we have today does not provide the mechanism for someone at a high level to look at the total menu—the things that could be done, perhaps should be done—and make decisions."

The allocation of resources has become a serious issue, Adams continued. To make sure we get the most from our expenditures and get it in the right directions, we need to establish some strategy—starting at the top, down through government organizations, universities, and industries.

A related issue, and another recurring conference theme identified by Adams, is technology transfer. The big problem is not the amount of technology we generate, it's getting our technology from the researcher to the developer to the factory. The United States still leads the world in basic scientific research, but it is losing its technological edge in the implementation process. "That's where the Japanese have beaten us to death," Adams said.

Companies that are successful in achieving innovation have facilities and programs dedicated to bringing it about,

Decision-makers from industry, government, and academia gathered to examine issues that influence a national agenda for R&D and US competitiveness in the world marketplace.

Adams noted, and several examples of this approach were described during the conference. These programs actively seek participation by technologists in setting up goals and programs and recognize the danger in attempting to manage research.

"Management interference with research can be deadly," Adams said, but industrial managers find this difficult to accept. They can manage strategy and funding and get the people, but then must "let them work." Management must also be prepared to accept failures, both in research and in high-risk development.

This, too, is difficult for old-style managers to accept, and Adams admitted that style was consistent with his own background in project management. The risks can, however, be mitigated by frequent reassessment—that is, by taking small steps, a policy exemplified and described by Boeing in one conference session.

But the major need, mentioned repeatedly during the conference, is for national leadership in establishing goals and priorities. An inhibiting factor cited by several speakers is the congressional budgeting process—separate budgeting, approval, and appropriation cycles for each program annually. The coordination and planning difficulties created by this process could be mitigated by elevating the science and technology advisor to cabinet level—a move recommended by Edward Donley, newly elected chairman of NASA's Commercial Development of Space Industry Advisory Group, in his opening-day address and echoed by conference attendees during small-group workshops on the following days.

Other recommendations included reinstating the R&D tax credit and focusing research through consortia and programs such as the Aerospace Industries Association's Key Technologies Program. That program, based on an industry consensus, has identified eight technologies—six of

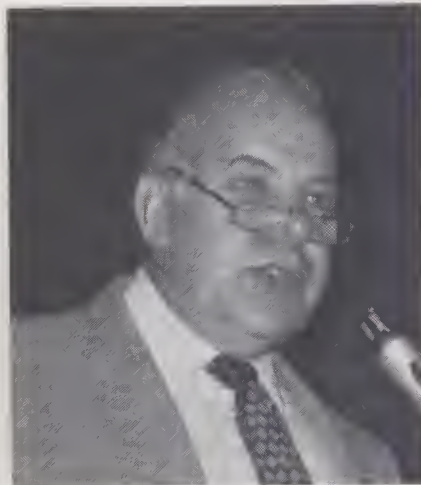
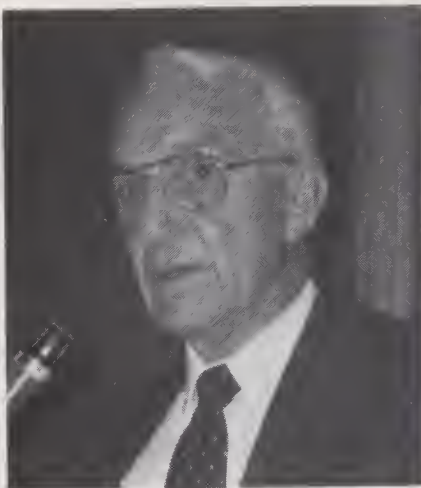
them computer-related—for concentrated attention. The goal is to double the productivity of available resources and, simultaneously, decrease by half the time it takes to bring a new technology to maturity.

The conference, called Stratrad for short, was held June 14-16 in Arlington, Virginia. It brought together nearly 500 industry, government, and academic leaders to assess ways of improving the strategic management of R&D, thus responding to the President's initiative on improved US competitiveness in the world marketplace. Stratrad was sponsored by the American Institute of Aeronautics and Astronautics, the Institute of Electrical and Electronics Engineers, the Electronic Industries Association, and the Aerospace Industries Association in cooperation with the US Department of Defense and the National Aeronautics and Space Administration. Adams, a past president of AIAA and retired chief executive officer of Martin Marietta, served as conference general chair; Gary A. DuBro, deputy director, Studies and Analysis Office of Naval Technology, chaired the Program Committee.

Focusing efforts to increase competitiveness

Federal budgeting process needs streamlining. To ensure its long-term competitiveness in a global economy, US industry must bring innovative, quality products to market more quickly. "In short," said Donna F. Tuttle, second in command at the Commerce Department and first speaker on the program at Stratrad, "the challenge before us is to live up to the theme of this conference and acknowledge that new approaches, techniques, processes, and management styles are essential to our long-term competitiveness."

From Commerce's perspective, the nation has been moving in the right direction. A recent departmental report identified emerging technologies, ones destined to be important in the year 2000, and pinpointed major barriers to their realization. Although intended for internal use, the report received an "intense response" from both the congressional and industrial communities, Tuttle said, indicating that "the country is literally thirsting for information and leadership aimed at regaining the competitiveness we took for granted for so long."



SPONSORS. Laurence J. Adams (left), past president of the American Institute of Aeronautics and Astronautics, was general chair of the National Conference on Strategic Management of Research and Development (Stratrad) held June 14-16 in Arlington, Virginia. IEEE President Russell C. Drew (right) chaired a session on successful implementation of R&D strategies. IEEE was a cosponsor of the conference, organized and operated by AIAA to address effective selection and achievement of top-level R&D goals. The program featured invited presentations by nationally known representatives from both the public and private sectors.

Tuttle went on to cite six statutes and initiatives that "have fostered federal, university, and private cooperation in such emerging areas." The Commerce Department has

(1) promoted use of R&D limited partnerships;

(2) spearheaded passage of the National Cooperative Research Act of 1984, which limited application of antitrust laws to permit research consortia (to date, 75 consortia have registered under this act);

(3) cleared the way for industry/national laboratory cooperation through the Federal Technology Transfer Act of 1986 and established the Office of Technology Transfer to make it work;

(4) negotiated for a series of bilateral agreements (in process, for example, are agreements with the USSR, Japan, Korea, India, and Indonesia);

(5) developed plans for shared production facilities; and

(6) proposed a new Department of Science and Technology to coordinate efforts within the Commerce Department.

The second keynote speaker, Representative Jack W. Buechner (R-Mo.), talked about US competitiveness from his perspective as a member of both the House Science, Space, and Technology Committee and the Budget Committee.

Buechner noted that the US has and will continue to have the edge in science and

engineering research, spending more on R&D per capita than any major competitor. But, he said, "Our lead is shrinking and may disappear."

"In 1970, we had twice as many scientists and engineers as Japan; today the numbers are roughly the same. In 1988, we'll spend roughly 2.7 percent of our gross national product on R&D—about the same as Japan or West Germany."

Although total R&D expenditures are substantial, the approach is "largely ad hoc and loosely coordinated." The US, he noted, is one of the few industrialized nations that does not have a single, national policy for the development and application of technology. The price we pay is the "inability to set and maintain priorities across broad fields of science from space to high-energy physics."

The only focal point for R&D is the President's Office of Science and Technology, and it lacks the authority to make choices and make them permanent. The Office of Budget and Management has the authority, but it is not organized to systematically consider trade-offs.

"When the administration's budget reaches Congress, it is split along arbitrary functions and then assigned to a wide array of authorizing and appropriating committees," Buechner said. This point was reemphasized later by Frank Press, president of the National Academy of Sciences, who

National standings in the coming information era are being decided now

The United States' loss of international computer markets is not caused just by restrictive trade practices or foreign competitiveness, according to Ben G. Matley and Thomas A. McDannold. Rather, these losses are part of a bigger picture in which foreign competitors, acting on behalf of their nations, meet the goals of national computer plans. These strategic plans are designed to leverage their countries into the information era of the 21st century, Matley and McDannold write in their recent book, *National Computer Policies*.^{*} Matley is a professor of mathematics and McDannold a professor of geography at Ventura College, Ventura, California.

The book points out that six countries, plus the European Economic Community, have produced official national computer plans: Japan, France, Great Britain, Taiwan, South Korea, and Singapore. Three countries are working on plans: Brazil, Australia, and Israel. But a number of countries important in computer manufacturing or applications do not have formal plans and from present indications do not intend to: United States, Canada, USSR, China, India, The Netherlands, Ireland, Sweden, and West Germany.

Japan. The Japanese national computer plan is the earliest, dating from 1971, and the broadest. It proposed the "realization of the information society" by the 21st century. Accomplishment of this goal, according to the Japanese plan, would bring about "a general flourishing of human intellectual creativity."

Raising the standards of human intellectual creativity could be achieved by educating people to fully appreciate and utilize computer products. The "development and settlement of a computer mind" within each individual was the sweeping goal.

The Japanese plan set forth three objectives, according to Matley and McDannold:

- The promotion of knowledge industries over smokestack industries.
- The development of solutions to the problems associated with smokestack industries.
- The focusing of effort on software solutions rather than hardware solutions.

Work on the Japanese plan has progressed ever since. For example, in 1981 the country embarked upon the well-known Fifth Generation project. In the mid-1980s emphasis shifted to software, according to the authors.

France. The French plan, published in 1978, also took a large view—the idea that France could both promote and control computerization so that it would serve "... the cause of democracy and human growth." The French also tended to go it alone, even inventing their own term, *telematique*. They felt that United States' dominance in computers, at that time, "was so pervasive that only French government action could assure the nation's desired telematic independence," Matley and McDannold report.

The Japanese and the French set up many projects, such as a national administrative database, use of computers in education, and videotex, that both provided a market for products and spread acquaintance with telematics throughout the population.

Britain and the EEC. The national computer plans of Great Britain and the European Economic Community tended to have less sweeping objectives, being focused on maintaining

or improving the economic position of the countries. In fact, they typically set up a governmental mechanism to sort out the enabling technologies and to finance research in them.

One of the favorite research areas is software technology. In the EEC program, this area was to be advanced by raising the level of the engineering standards that underlie software development. The effort was to be three-pronged, Matley and McDannold state, "first involving formal work in areas such as mathematics, taxonomy, software metrics, and modeling."

The second thrust "would deal with software engineering practices and seek to apply the findings of research to industrial settings. It would include work on software life cycles, requirements specifications, design techniques, verification, validation, and life maintenance of software. In addition, methods and tools would be developed for software production."

The third prong would address software as an economic product in the marketplace.

Other frequently mentioned enabling technologies were knowledge-based or expert systems, man-machine interfaces, very large scale integration, parallel processing, and computer-integrated manufacturing.

Pacific Rim and Sweden. The plans of the Pacific Rim countries—South Korea, Taiwan, and Singapore—tended to be more immediately practical. They focused on identifying specific hardware or software products that could be produced in the near future.

Sweden's absence from the list of countries with national computer plans may surprise some. It has laws on data privacy and transborder data flow; it has funded development of VLSI production and participated in a Nordic States telecommunications satellite project. But it has not felt the need for a formal computer plan.

United States. As for the United States, Matley and McDannold assert that it "needs a national computer policy [to] coordinate and strengthen its national computing effort." They regard a national computer plan as necessary to offset the country's declining trade balance in electronic products. They cite the idea that "no one nation, certainly no one company, [can] any longer afford to do everything in the manner of the 1960s when research and development was redundant and competitive."

They recommend the establishment of a National Computer Coordination Council at the federal level with representatives from the government agencies concerned and from all computer-related organizations—academic institutions, professional societies, and industry. The council's responsibilities would include setting national priorities in the technical and social aspects of computing. In effect, it would develop, then administer, the national computer plan.

The authors are aware of a certain "lack of national interest" in such a plan. They feel this lack is "rooted in the view that no good comes from government planning."

If information industries are to be to the 21st century what smokestack industries were to the 20th century and if we are living now in the transition period, then this American outlook is "most unfortunate and shortsighted," the authors conclude. "A national computer plan could give us the long-term view that is needed to help the country through this transition period."

—Ware Myers

^{*}Ben G. Matley and Thomas A. McDannold, *National Computer Policies*, Computer Society Press, Catalog Number 784, IEEE Computer Society, Los Alamitos, Calif., 1987, 172 pp.

noted that funds for science and technology are requested by 15 federal departments and agencies without coordination, go through 14 separate budget functions spread over at least six divisions of the Office of Management and Budget, and require approval of nine appropriation committees in Congress.

"It is astonishing but true," Press said, "that nowhere in the federal budget-making process is there an evaluation of the complete federal budget for science and technology and the overall rationale in terms of national goals."

Both speakers offered solutions within the existing structure.

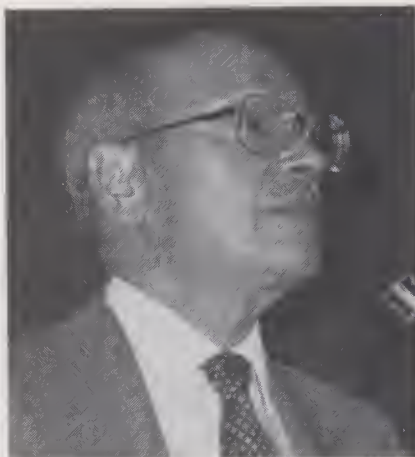
Buechner recommended a serious study to determine better organization for both the legislative and executive branches. Press advocated a coordinating role within the executive branch for the President's science advisor, who would assume responsibility for providing a rationale for a coherent and adequate federal science and technology budget. The function would be carried out as an instrument of Presidential policy and a resource to OMB, and the science advisor would be named a member of the cabinet without portfolio. Press also recommended that the Budget Committee provide Congress with a cross-cutting evaluation. Another possibility would be a new joint committee consisting of representatives from all the authorization and appropriations committees that deal with science and technology budgets.

Buechner concluded that it all came down to personal responsibility. Although the words "science" and "technology" may seem to convey remoteness and machine-like impersonality, they are "intensely human activities." Success, he told the audience, will depend on "your vision and your efforts."

In a similar vein, Press quoted Damon Runyan as saying, "The race is not always to the swift and the strong, but that's the way to bet." Press is betting on the US.

Aerospace industry targets key technologies. Six of the eight key technologies selected by the Aerospace Industries Association as essential to the further development of the aerospace industry overlap the computer industry. They are

- very-large-scale integrated circuits,
- advanced software development,
- advanced sensors,
- optical information processing,
- artificial intelligence, and



KEYNOTERS. As the leading nation in scientific research, the United States has a comparative advantage that it must hold and exploit, according to Frank Press (above), president of the National Academy of Sciences. Marginal increases in funding would yield "enormous returns," he said. To focus R&D efforts, Press advocated a cross-cutting review of government programs and a coordinating role for the President's science advisor. Similar recommendations were made by Congressman Jack W. Buechner (above right) and Senator Jeff Bingaman (below right), who also spoke at Stratrad.



• ultrareliable electronic systems. The other two technologies are advanced composite structural materials and rocket and air-breathing propulsion systems.

The AIA Technical Council concluded that "a strong and growing US aerospace posture can be ensured only by more aggressive industry, government, and university technology programs," according to John M. Swihart, vice president, international affairs, Boeing Company, who is also chairman of the Key Technology Steering Group of the Technical Council.

The council, with the assistance of senior technologists from AIA member companies, reviewed over 100 technologies, seeking those with high leverage and risk but with the potential of exceptional long-term payoff and broad application. The industry reached consensus on these eight technologies.

The AIA proposes that these technologies "be given developmental priority on a national basis," Don Fuqua, president of the association, reported to the conference. To carry out this program, the AIA has established a Technology Policy

Forum on which top-level government officials, industry executives, and university representatives are serving. It held its first meeting in July 1988.

"I commend the AIA for its effort to develop a road map of key technologies for the 1990s," Senator Jeff Bingaman (D-NM), chairman of the Defense Industry and Technology Subcommittee of the Senate Armed Services Committee, told the conference. "Following up on that example, I authored an amendment on the fiscal year 1989 Defense Authorization Bill which will require DoD and DoE to come up with a plan for developing the 20 technologies they consider most essential to ensuring the long-term qualitative superiority of US weapons systems."

Managing the R&D process in computers and electronics

The R&D consortium approach is working. "After five years of operation, we

The superconductivity race

Are we in a race to practical superconductor applications? Or are we still in the research stage? At least five Stratrad speakers touched on these issues. The superconductivity case is an excellent example of the difficulty of applying strategic management to the messy, complicated, and often unknown facts of the real world.

At first there was no race. That was back in 1911 when Heike Kamerlingh Omnes discovered the phenomenon at a temperature of 3 Kelvin. It was too hard at that time to get down that close to absolute zero for superconductivity to have any practical applications.

In the 1950s the race warmed up a little. A class of materials was discovered that became superconducting at about 20-K. Unfortunately, that still required liquid helium cooling (at 4-K) and most applications remained uneconomic.

Then, beginning in April 1986, one or another group of scientists worldwide announced new materials almost every month. The critical temperature (at which the material becomes superconducting) rose from 30-K to about 125-K over this two-year period. There is good reason now to believe that the critical temperature will rise still further, said R.P. Caren, vice president for science and engineering, Lockheed Corporation.

So the research race is in full swing, but the applications race is still awaiting the arrival of a practical material.

The operating temperature of liquid nitrogen—a much more practical coolant than liquid helium—is 77-K. The temperature in deep space is about the same, implying that devices in space could operate without expensive and heavy cooling systems. But superconducting materials seem to operate best at a temperature about half the critical temperature. So it would be nice to get the critical temperature up to about 150-K.

There are other difficulties, too—the subject rapidly gets very complex. (Refer to the May 1988 issue of *IEEE Spectrum* for a special report on the subject.) Much more research is needed. "We don't know nearly enough about the characteristics of these materials," said Alex Ignatiev, director of the Space Vacuum Epitaxy Center at the University of Houston, who filled in for Paul Chu, widely famed for research in this field.

Thin-film applications will be the first field in which competition blossoms. A film of superconducting material on a substrate can be used for electronic interconnections, sensors, and switching devices.

"In the microelectronic environment and small-device applications, the expectation is that quite soon—in the less-than-five-years time frame, and maybe even much less than that—we will see some progress," Ignatiev said.

"In something less than six years from now, there are going to be all kinds of products based on superconductivity," said Gerald P. Dinneen, corporate vice president, science and technology, Honeywell, based on his view that competition is forcing development into a faster-than-six-years time frame.

The Strategic Defense Initiative Organization has embarked on \$13 million worth of small projects to build devices and sensors based on superconductivity, Lt. Gen. James A. Abrahamson, SDIO director, stated. He expects them to come to actual applications in about two years. If they can be moved up the temperature scale to levels above space temperature, the spacecraft won't have to carry cryogenic equipment. The implications for spacecraft design and affordability are profound, he said.

Only Japan's Sharp Corporation actually has a product on the market. It is a sensitive magnetic-field measuring device. "It is kind of strange," Ignatiev said, "at a time when we have

all been trying to get the best performance out of our materials and generate the best materials available, that Sharp has taken what there is already and said, 'how can we use it?'"

As a matter of fact, what is needed for this application is not an ideal superconductor where the resistivity curve drops precipitously to zero, according to Ignatiev. What it requires is a material where the curve drops gradually to zero.

Competition in large-scale machinery applications, such as power generation, transmission, and storage, is "much more distant," according to Ignatiev. "You first have to solve the problems of mechanical strength of the superconducting materials. You have to get the current density in these bulk materials [in contrast to thin-film materials] high enough that they, in fact, do better than copper. Current bulk materials have current densities on the order of 2,000 amperes per square centimeter. Copper is up in the 50,000 to 60,000 ampere-per-square-centimeter range. So better use copper."

These applications are already a gleam in the eye of Abrahamson. "For us the drama in this high-temperature superconductivity revolution is what it can do to magnetics. Obviously it can be applied to such things as particle beams, power systems, and storing large amounts of power in inductance for free-electron lasers."

Given the current state of research in this field, the strategic question is how much effort high-technology players should put into it. At the first of this year, the Microelectronics and Computer Technology consortium started a phase-one effort to study what the impact will be on the computer, semiconductor, and aerospace industries.

"As far as I know, we at MCC have the only industry-sponsored consortium effort in the field," Grant A. Dove, MCC chairman and chief executive officer, said. "We have 13 companies funding it, but at a very low rate—\$1.3 million."

Dove found that American industry was a bit cynical in this area as it spent a lot of money in the past without too much payoff, for instance, in the Josephson junction area. Some of the people he talked to think that practical implementation may be quite far off. So many of them are sitting on the sidelines.

By contrast, he continued, the Japanese have started up at least six efforts. One of them has an up-front commitment of at least \$30 million from 42 companies with an ongoing commitment of something like \$8 million or \$10 million a year. It seems as if the Japanese can form a consortium in days or weeks, he said.

"We are wondering if these materials will work," Dove concluded. "They are spending money to see how this technology will work."

The wonder is, said Dinneen, continuing the theme, "how much money we should be spending on superconductors when, in fact, we can't see the market. Given that critical temperatures are now 100 to 120 Kelvin, we still don't see practical applications."

Lockheed now has about 50 people working on superconductivity research, but Caren feels that the effort is still subcritical. "The fact of the matter is, as I look at the national effort, I believe it is subcritical, too. It is subcritical in the way it is organized. We don't really have, in my mind, a national road map. It is not clear that we have settled the application priorities."

The problem is not so much in the actual research currently in progress. The problem is, when somebody finds a material with the right kind of characteristics, how rapidly can we rush that into production and make the material available, Caren said. "We need a reservoir of money so that when the time comes and the breakthrough occurs, we can act in a hurry."

—Ware Myers

have shown that the R&D consortium approach works," reported Grant A. Dove, chairman and chief executive officer of Microelectronics and Computer Technology Corporation (MCC) for the last year. "The whole idea of cooperative joint research is to conserve resources, to get leverage at the appropriate phase of R&D for appropriate kinds of projects.

"Those of us with experience in large corporations know that we never have enough R&D money to fund all the good ideas," Dove recalled. "Also, in many cases you don't start something because you are short of the critical skills."

MCC's 1988 budget is about \$70 million; it has 19 member companies and 440 employees. Its research results have proven to be generic—that is, the members have been able to use the results either in their own development or as new products or services under their own trademarks.

"Still, it is not an easy job," Dove said. "At times I have had a little more 'fun' than I anticipated."

Joint research in the United States is bigger than most people realize. Dove cited Bell Communications Research, formed by seven "baby Bell" operating companies; Sematech, intended to improve the competitiveness of the US semiconductor industry; the Semiconductor Research Institute, distributing industry funds to universities; the Software Productivity Consortium of 14 aerospace companies; and MCC.

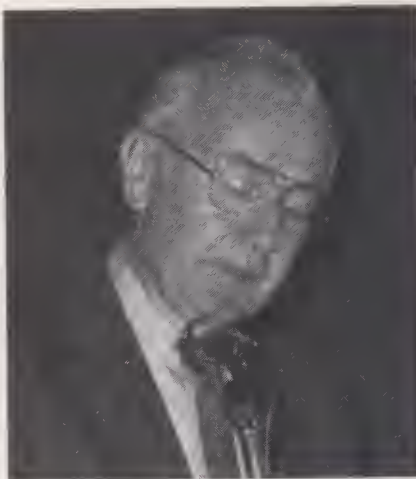
These consortia, just in the computer and electronic industries, add up to around \$600 million a year. With the exception of \$100 million in DARPA funding for Sematech, essentially all the rest of the money comes from industry.

"At MCC," Dove continued, "the average investment of the members is about one percent of their R&D expenditures."

Still, the Japanese and the western Europeans are doing much more cooperative research. "We know of at least 120 cooperative R&D projects in Japan," Dove said. They are spending in the \$1.5 billion range.

"In western Europe we can identify some 400 cooperative research projects with a total budget around \$2 billion," he added. "We need more of it. The research-consortium idea was invented to leverage people, money, and time."

Shorten the time from idea to product. Competition is making it necessary to get new products to the marketplace faster,



PANELISTS. Grant A. Dove (above), MCC chief executive officer, chaired a Stratrad panel on managing the R&D process. During his 37 years in the industry, Dove said, he has been involved in five generations of computers and has seen the traditional sequence of R&D management replaced by "synchrony," in which the CEO heads a flatter organization and functions "more like the conductor of a symphony." The other panelists were Gerald P. Dinneen (above right) of Honeywell, Lloyd M. Thorndyke of ETA Systems (below right), and Howard L. Yudkin (not shown).



speaker after speaker emphasized. It took Faraday's ideas half a century to get from his laboratory to the nascent electric power industry of the 1880s, Gerald P. Dinneen, Honeywell corporate vice president for science and technology, pointed out.

Recently a sensor based on new technology went from concept to market in six years, but the responsible vice president told Dinneen, "That is too long. We can't afford to take that long, because our competitors will not let us have that leisure."

ETA Systems is getting its supercomputers to market sooner by substituting simulation for prototypes. "We go immediately from simulation to production," said Lloyd M. Thorndyke, chairman. "There is no place in between."

There is no place for the chief executive to hide either. "It takes a fair amount of intestinal fortitude," Thorndyke admitted. The simulations are run by young engineers, perhaps two years out of college.

"My career is linked to these young people," Thorndyke went on. "They are mak-

ing what amount to \$10-to-\$50-million decisions."

ETA Systems started with 127 people; it has 900 today, 400 of them professionals.

Thorndyke believes that "you can't generally develop high-risk programs [like ETA's supercomputers] in a large corporation." In that situation the chief executive officer has to approve things that are happening four levels below him that he doesn't know about. The small dedicated group is the way to go. That way the young simulation experts are not so far away organizationally.

Capitalize software development. Doing something about the software crisis creates a dilemma for management, according to Howard L. Yudkin, president and chief executive officer of the Software Productivity Consortium. To get the payoff, organizations will have to capitalize the software development process.

"God forbid that you should capitalize

the software development process," Yudkin exclaimed. A software engineer costs about \$120,000 a year, including organizational overhead. "That I should spend \$30,000 every five years to make that engineer useful is just totally unacceptable to either government or industrial management."

Of course, Yudkin was exaggerating for the sake of emphasis. Some organizations are making the investment in the software development process and are reaping rewards. Most are not.

The problem lies in recovering the cost of this investment. Government won't like it if contractors try to recover the investment. More than that, if an organization wants to exploit reuse, it would have to capitalize and invest in reusable software. "You don't have the mechanism by which you can capture recovery of that," Yudkin said.

The underlying problem is that the software development process is hard to understand. It is particularly hard for managers who are divorced from software technology. Yudkin called software "ethereal." Consequently, managers have a lot of trouble dealing with the software process.

One reason is that they can't picture it, as they can a physical product in a drawing or a prototype, Yudkin said. For that reason they have trouble relating to it. It is hard to pinpoint where the problem is. It is hard to visualize how to fix it. Most managers find it easier to analyze a hardware problem.

To make matters worse, their staffs are not literate in the subject either. We have the spectacle of staff who do not understand trying to interpret to managers who do not understand. "Naturally, it doesn't

work out very well!" Yudkin said.

Beyond that, what Yudkin called "MBA analysis" is looking for a payoff in three years, and that makes the problems of researching the software process even worse.

One of the results of this frustration, particularly in government, has been regulations dictating how software should be developed rather than an effort to understand the process. These regulations have legislated against "really terrible practice," Yudkin said, but have limited the ability of developers to adopt "good practice."

One of the concepts in the regulations was the waterfall process of developing software: first, you do the requirements, then design, code, test, etc. It was a good idea at the time, in the 1970s, but it was an idealization of what actually goes on—as if it were possible to write a meaningful requirement in the absence of understanding the implications of what would grow out of actually building it.

"This was an interesting idealization—that [the authors of] government regulations actually believed that we as professionals could do this," Yudkin said. "This form of regulation is actually holding back advances in software development and methodology."

Fortunately, at least 14 defense-industry companies have realized that they have to do something about the software problem and they have turned to the consortium concept, establishing the Software Productivity Consortium. "I hope we can stand up to the challenge," Yudkin said.

The usual answer is to automate the steps of the software process. "If you analyze it, however, there is not a lot that goes beyond the automated systems that

already exist," Yudkin pointed out. "That says you have to do something different, not just automate the steps that you already have."

There seem to be three major approaches, he said. One is the attempt to automate bigger steps in the individual activities.

Second is the hope to embed some kind of smarts in the development environment. This could be very attractive, but you have to have the smarts first.

Third is the possibility of changing the process. Maybe the waterfall concept was wrong. "Maybe we ought to go back and look at the classical hardware development ideas," Yudkin went on. "We could start with some experimentation and build that into the process. We could rethink the process and then automate that."

As soon as you begin to think along these lines, some interesting points come out. "In order to prototype graphically and efficiently, you have to prototype something that existed before," Yudkin explained. "That introduces the idea of reuse, and that seems to be where most of the payoff is."

Getting new technology into the manufacturing/development cycle

Transferring technology to existing product lines. "The process of going from one bit on a chip to one million bits on a chip has been one of incremental, or cyclical, or iterated improvement," said Ralph E. Gomory, IBM senior vice president for science and technology. "It is in this process of incremental improvement or cyclical development in which most of the competitiveness issue is fought out."

In general, the lack of competitiveness in American industry has not been in the introduction of entirely novel products. It has been in the day-to-day competition over existing products.

"It is this process—getting new technology from the laboratory into the evolving product—that we need to focus on," Gomory urged, introducing a panel of three speakers.

The technology supermarket. "Each time the development cycle goes around, the design team has an opportunity to employ new technologies," said Philip M. Condit, executive vice president, Boeing Commercial Airplane Company. At that

Strategic management of R&D: A literature search

A 56-page literature search, compiled by Carol K. Sterkin of the Jet Propulsion Laboratory of the California Institute of Technology, is an advance product of the Stratrad Postconference Subcommittee. According to Sterkin, it contains a major portion of what is available on strategic management.

The document provides half-page abstracts of reports, journal articles, conference papers, and books on strategic planning and management of research and development. It also provides the sources from which the literature can be obtained.

A copy can be obtained from the National Technical Information Service, 5285 Port Royal Rd., Springfield, VA 22161, phone (703) 487-4600, for \$14.95 plus \$3.00 for shipping and handling. Order by number N88-14847.

point they go shopping—looking through technical journals, contacting experts, reading reports, picking the brains of suppliers, and so on. This transfer is a “pull” process—the designer is looking for things he can use on the next cycle, as contrasted with the “push” process, where researchers try to reach users with new technology.

“What we have in technology transfer is fundamentally a marketing problem,” Condit went on. “The engineer is shopping at a technological supermarket.”

The researcher can put his output on an eye-level shelf where the designer can easily see it, or he can let it languish on a bottom shelf where it will not be seen. “My view is that the role of the researcher must contain that marketing element,” Condit said.

If the research results are seen by many designers, they will find many uses for it that the originator never thought of. New technology will be incorporated in products more rapidly, Condit said.

Short development cycles. “It is extremely important that a technology product be brought to the market in a timely fashion,” said Robert Stratton, vice president, corporate staff, and director, Central Research Laboratories, Texas Instruments. “Studies show that a six-month slip in new product introduction can reduce the life-cycle profits by about one-third. An interesting point is that an overrun of the development costs by as much as 50 percent has less than a 4 percent impact on profits. In studies of product introduction we have found that an n -month slip in introduction time resulted in n times a hundred million dollars change in profits.”

Gomory further emphasized this point. “If you have two companies, drawing on the same set of ideas, and one has a two-year product-development cycle and the other has a year- and-a-quarter cycle, the one with the shorter cycle will appear to be the technical innovator.” Differences in cycle length of this size, and even larger ones, do exist, he said.

The shorter cycle gets the new ideas to market sooner. Moreover, that advantage, when iterated through a few cycles, becomes larger and larger.

Technology transfer from government laboratories. It is difficult to transfer new technology from a government laboratory or from a government R&D sector to potential users, according to Robert R. Barthelemy, now program manager for

the National Aerospace Plane Joint Program Office, drawing upon his recent experience as technical director of the Air Force Wright Aeronautical Laboratories at Wright-Patterson Air Force Base. The labs have a budget of about \$1 billion a year.

“It is difficult because we made it that way,” Barthelemy said. “We have put in motivational systems, funding structures, and career paths that have absolutely nothing to do with technology transfer.”

Government R&D expenditure runs into tens of billions of dollars per year, he went on. If we are not transferring the results of this effort to other parts of the economy, including the commercial and industrial sectors, then “there is an awful

lot of technology going down the drain.”

Token incentives have been proposed and implemented in the last few years, but they are not doing the job. “We have to set up mechanisms to do it,” he urged.

He sees some difficulties. For instance, one must be careful not to detract from the innovative environment that is created by leaving basic researchers and exploratory developers alone.

“The only solution I can offer,” he concluded, “is to say, let’s manage the process. There have to be goals, plans, and organizations. There have to be means for transferring people.” The government has to decide that technology transfer is one of its primary missions and set about providing the means to make it work. □

Challenge...

The Future

For over four decades, Los Alamos National Laboratory has challenged the frontiers of science, researching an exciting range of phenomena. Pioneering men and women of science have accomplished breakthrough discoveries in many areas, combining basic sciences with engineering disciplines and technological advances. As one of the largest multidisciplinary, multiprogram national laboratories in the United States, we are also internationally recognized as one of the most prestigious scientific institutions in the world.

Computer Security Professional

The DOE Center for Computer Security at Los Alamos currently seeks computer professionals to participate as a team member supporting the Department of Energy's computer security program. Tasks include a broad range of R&D projects related to the security of DOE computer systems. Current projects include security model development, software verification, expert systems as applied to the security of networks and anomaly detection, and computer security education.

This position requires an in-depth knowledge of one or more of the following: computer security, modern operating systems or computer networking. A background in languages including 'C', PASCAL or FORTRAN is required. Ability to work in a team environment is needed. A Master's, PhD or equivalent combination of education and experience in computer science, engineering or mathematics is needed.

Superior compensation and benefits are provided. For prompt consideration, send resume to: Jane Roberson (MS P280), Personnel Services Division 80207-BJ, Los Alamos National Laboratory, Los Alamos, NM 87545.

Affirmative Action/Equal Opportunity Employer
U.S. Citizenship Required

University of California

Los Alamos



Sponsored by:
IEEE/CS TC on Data Engineering &
ACM SIG on Computer Architecture
In Cooperation with:
IEEE/CS TC on Distributed Processing
& INRIA

ADVANCE PROGRAM International Symposium on Databases in Parallel and Distributed Systems

December 5-7, 1988

Austin, Texas

Symposium General Chair

Joseph E. Urban
University of Miami

Co-Program Chair

Sushil Jajodia
George Mason U.
Won Kim
MCC
Avi Silberschatz
UT-Austin

Program Committee

Rakesh Agrawal, AT&T
Bell Labs
Francois Bancilhon, INRIA
John Carlis, U. of
Minnesota
Doug DeGroot, TI
C. Ellis, Duke U.
Shinya Fushimi, Mitsubishi
H. Garcia-Molina,
Princeton U.
Theo Haerder, Germany
Yahiko Kambayashi,
Kyushu U.
Gerald Karam, Carleton U.
Michael Kifer, SUNY-
Stony Brook
Roger King, U. of Colorado
Hank Korth, UT-Austin
Ravi Krishnamurthy, MCC
Duncan Lawrie, U. of
Illinois
Witold Litwin, INRIA
Eliot Moss, U. of Mass.
Erich J. Neuhold, iGMD
Anil Nigam, IBM Thomas J.
Watson Lab.
N. Roussopoulos, U of MD
Sunil Sarin, CCA
Y. Sagiv, Hebrew U.
G. Schlageter, Fern
Universitat
Jim Smith, ONR
Ralph F. Wichter, ONR
Ouri Wolfson, Technion
Clement Yu, UI-Chicago
Stan Zdonik, Brown U.

Publicity Chair

Ahmed K. Elmagarmid,
Purdue U.

Tutorials Chair

M. Tamer Ozsu, U. of
Alberta

Local Arrangement

Hong-Tai Chou, MCC

Finance Chair

Edward T. Lee, U. of
Miami

December 5, 1st Day

2 Tutorials 8:30 - 12:00

Tutorial 1: Parallel Computing and Database Management, *Bruce K. Hillyer*

Tutorial 2: Integrating Heterogeneous Distributed Databases: Requirements, Concepts and Solutions, *Ahmed K. Elmagarmid and Amit P. Sheth*

Opening Remarks 1:00 - 1:15: *Joseph Urban, University of Miami, Florida*

1:15-1:30: *Won Kim, MCC*

Keynote Address 1:30 - 2:15: *J.C. Browne - University of Texas at Austin*

Session 1 2:30 - 3:45 Object-Oriented Systems

Chair: *Gerald Karam, Carleton University*

"The Relation Between Problems and Large-Scale Concurrent Systems and Distributed Databases (invited paper)," *Gul Agha, Yale University*

"Writing Reliable Servers: Design and Implementation in Avalon/C++," *Richard Allen Lerner, Carnegie Mellon University*

"Performance Modeling of Distributed Object-Oriented Database Systems," *Jeffrey A. Brumfield, University of Texas at Austin, Janet Miller, Hong-Tai Chou, MC*

Break

Session 2 4:00 - 5:30 Logic-Based Systems

Chair: *Ravi Krishnamurthy, MCC*

"Exploiting Concurrency in a DBMS Implementation for Production Systems," *Louisa Raschid, Timos Sellis, Chih-Chen Lin, University of Maryland*

"Sharing the Load of Logic-Program Evaluation," *Ouri Wolfson, The Technion*

"Multiprocessor Transitive Closure Algorithms," *Rakesh Agrawal, H.V. Jagadish, AT&T Bell Laboratories*

Reception 6:00 - 9:00

December 6, 2nd Day

Session 3 9:00 - 10:30 Parallel Database Systems

Chair: *Rakesh Agrawal, AT&T Bell Laboratories*

"Parallelism in Bubba (invited paper)," *Haran Boral, MCC*

"Parallelizing FAD, a Database Programming Language," *Brian Hart, Scott Danforth, Patrick Valdurez, MCC*

"JAS: A Parallel VLSI Architecture for Unformatted Data Processing," *K.C. Lee, O. Frieder, V. Mak, Bell Communications Research*

Break

Session 4 11:00 - 12:30 Parallel Join

Chair: *Anil Nigam - IBM Thomas J. Watson Lab.*

"Parallel Join Algorithms on a Network of Workstations," *Xiao Wang W.S. Luk, Simon Fraser University*

"A Robust Protocol for Parallel Join Operations in Distributed Databases," *S. Bandyopadhyay, A. Sengupta, University of Windsor*

"Effect of Skew on Join Performance in Parallel Architectures," *M. Seetha Lakshmi, Philip S. Yu, IBM Thomas J. Watson Lab.*

Session 5 2:00 - 3:30

Panel: Chair: *John Carlis, University of Minnesota*

"Parallelism in Databases: What, Why and Whither"

Break

Session 6 4:00 - 5:30 Distributed Query Processing

Chair: *Hong-Tai Chou - MCC*

"A Case Study for Distributed Query Processing," *P. Agrawal, D. Bitton, K. Guh, C. Liu, C. Yu, University of Illinois at Chicago*

"Parallelism in Processing Queries on Complex Objects," *T. Haerder, H. Schoning, A. Sikeler, University of Kaiserslautern*

"Heuristic Algorithms for Distributed Query Processing," *P. Bodorik, Technical University of Nova Scotia, J.S. Riordon, Carleton University*

Reception 6:00 - 9:00

December 7, 3rd Day

Session 7 9:00 - 10:30 Transaction Processing

Chair: *Sunil Sarin, Computer Corp. of America*

"Node Autonomy in Distributed Systems (invited paper)," *Hector Garcia-Molina, Boris Kogan, Princeton University*

"Performance Evaluation of Global Reading of Entire Databases," *Calton Pu, Christine H. Hong, Jae M. Wha, Columbia University*

"Multiprocessor Main Memory Transaction Processing," *Kai Li, Jeffrey F. Naughton, Princeton University*

Break

Session 8 11:00 - 12:30 Distributed Databases

Chair: *Ophir Frieder, Bell Communications Research*

"VIP-MDBS: A Logic Multidatabase System," *Eva Kuhn, Thoms Ludwig, Technische Universitat, Wien, Austria*

"Integrating Relational Databases with Support for Updates," *M. Samy Gamal-Edin, Clarkson University, Gomer Thomas, Bell Communications Research, Ramez Elmasri, University of Houston*

"Robust Transaction Routing in Distributed Database Systems," *Yann-Hang Lee, Philip S. Yu, IBM Thomas J. Watson Lab*

Tutorial #1: Parallel Computing and Database Management,

Instructor: *Bruce K. Hillyer, AT&T Bell Labs*

- Outline:
- Cost and performance trends in computer hardware
 - Issues and approaches in parallel processing
 - Architectures for parallel non-numeric computing
 - Parallel algorithms for relational database management
 - Proposed parallel database machines

About the Instructor: Dr. Bruce K. Hillyer received the Ph.D. in Computer Science from Columbia University in 1986, and is a member of technical staff at AT&T Bell Laboratories. He has published in the areas of parallel relational database management and parallel execution of AI production systems. His present research interests include parallel computation, disk I/O in parallel systems, and large-scale parallel data management. Dr. Hillyer is a member of the ACM and IEEE Computer Society.

Tutorial #2: Integrating Heterogeneous Distributed Databases: Requirements, Concepts and Solutions,

Instructors: *Ahmed K. Elmagarmid and Amit P. Sheth*

Outline: In this tutorial, we will look at the issues related to integrating heterogeneous and distributed databases. Many organizations today are faced with a need to deal with heterogeneous distributed databases because of decentralized control, benefits of distribution and historical reasons. They need ways to have evolutionary database expansion to promote data sharing among users and applications, and easier application development. These needs can be satisfied by system and database integration. In this tutorial, we will study: The requirements for integrating heterogeneous databases, the architectural alternatives for an integrated database, the concepts and basic technologies required to achieve integration at logical and physical levels, and case studies of many prototype and recent commercial systems.

About the Instructors:

Professor Ahmed K. Elmagarmid is an associate professor of Computer Science at Purdue University. His research interests are in Parallel and Distributed systems. In the last three years he has been concentrating on heterogeneity issues in distributed databases. Before joining Purdue University, he was an assistant professor of Computer Engineering at Pennsylvania State University. He received his M.S. and Ph.D. in Computer & Information Science from the Ohio State University, Columbus, Ohio in 1985.

Dr. Amit P. Sheth is a Staff Research Scientist at the Systems Development Group of UNISYS. He has led a project on developing heterogeneous distributed database system for a manufacturing environment. He has also performed research and development of federated database tools including tools for schema integration and view update. Currently he is working on database design and integration, and integration of AI systems with database systems.

REGISTRATION FORM

Complete and return this form with your check made payable to ISDPDS and send it to: Professor Joseph Urban

University of Miami
College of Engineering
P.O. Box 248294
Coral Gables, FL 33124

Advance Registration (prior to November 4, 1988)

	Member	Non-Member	Full-time Student
Tutorial	\$90	\$110	\$90/\$110
Symposium	\$205	\$255	\$40

Late Registration (after November 4, 1988)

	Member	Non-Member	Full-time Student
Tutorial	\$110	\$135	\$110/\$135
Symposium	\$240	\$255	\$40

☐ Check enclosed ☐ Visa ☐ MC ☐ Amex

Credit Card No. _____ Exp. Date _____

Signature _____

Total Amount Enclosed: _____

IEEE/ACM Membership Number: _____

Name: _____

Organization: _____

Street Address: _____

City/State/Zip: _____

Country: _____

HOTEL REGISTRATION FORM

Group: International Symposium on Databases in Parallel and Distributed Systems

Reservation requests must be received by November 21, 1988.

Name _____

Company/Affiliation _____

Address _____

City _____ State or Province _____

Zip, Country _____ Telephone () _____

Sharing Room With _____

Single (\$55) or Double (\$55) _____

Arrival Date _____ Departure Date _____

To guarantee your hotel reservation include credit card information, or a check for one night's deposit made payable to: Double Tree Hotel. You will be charged for the first night if reservations are not canceled 24 hours prior to arrival.

Credit Card Type _____
(AMEX, DINERS CLUB, VISA, MC, CARTE BLANCHE)

Credit Card No. _____ Exp. Date _____

Mail deposit and completed hotel reservation form to:

Doubletree Hotel, 6505 IH-35 North, Austin, TX 78752

Attn: Reservations

STANDARDS

Editor: Helen M. Wood, National Oceanic and Atmospheric Administration, Rm. 1069, Federal Bldg. No. 4, Washington, DC 20233, phone (301) 763-1564.

Computer community urged to pursue supercomputer benchmarking

Joanne L. Martin, IBM Research

John P. Riganati, Supercomputing Research Center

Performance measurement, particularly as it applies to supercomputing systems, has been approached through a number of ad hoc and experiential processes. Although each of these approaches attacks some aspect of the problem, no substantive theory has been developed to serve as the framework for scientific judgments.

The ad hoc procedures are especially troublesome in that they may give misleading results if applied in an environment that is not similar to the one in which they evolved.

Our purpose here is to put these approaches into perspective and to describe and support a process, centered at the US National Bureau of Standards Institute for Computer Science and Technology, which has the potential of bringing some unity to the current disarray.

This problem was stated concisely in "Performance Measurement"¹:

The need for a performance measurement of complex computing systems arises from a number of disparate sources. The result is a set of requirements which are incomplete and inconsistent. Comparison of different . . . systems at the user level, evaluation of new architectural ideas at the research level, and separation at the developmental level of the effects of the compiling process from the effects of hardware architecture all would greatly benefit from a set of measurements, theoretical or practical, which have predictive validity.

The use of an analogy may at least foster a partial understanding of what is needed. For linear time-invariant systems, knowledge of the system's response to an impulse-forcing function allows calculation of the system's response to any arbitrary function by convolving the impulse response with the arbitrary function.

An analogous "describing function" for nonlinear, time varying, complex

computing systems (which allows calculation of the behavior of the system for arbitrary inputs) is the holy grail that all benchmarking approaches—consciously or unconsciously—seek.

Far from having such a describing function, many of the performance-evaluation results that appear regularly in the literature suffer from an incomplete description of the conditions present during the measuring. Furthermore, terminology usage is frequently inconsistent, even within a single paper.

After a brief description of the benchmarking process, we will describe and support one approach—begun at the NBS in 1985—that we suggest is the best hope for a near-term process to solve some of these difficulties. Because of the nature of the approach, it is most appropriate that other laboratories participate as peer entities, with the NBS serving as the coordinating focus.

The benchmarking process. The term *benchmark* comes from its use as a surveyor's mark made on a stationary object of previously determined position and elevation. This mark is used as a reference point in tidal observations and surveys.

Computer benchmarks may be viewed the same way. They are starting points in the performance evaluation process. As such, they must be widely accepted and, hence, be treated as "standards," especially if the inevitable enhancements characteristic of our dynamically evolving field are to be quantitatively appreciated.

The benchmarks themselves provide a means for assessing base-performance statistics across widely different systems. By themselves, these base-performance statistics are not sufficient to evaluate a system, but they serve as an initial zeroth-order common term for such evaluation.

Many pseudo benchmarks have evolved. Common names associated with benchmarking are the Gibson Mix, Whetstones, Dhrystones, the Livermore Loops, the Linpack routines, the Los Alamos Benchmark set, and the NASA Ames kernels.

In addition, every manufacturer of computer systems has evolved a set of benchmarks, often considered proprietary and therefore not shared; manufacturers use them as bases for internal critiques of evolving system implementations.

Almost always, performance figures based on these proprietary measurements only state performance relative to previous incarnations of the same architecture.

In addition, specific industries use a number of applications. For example, the Society of Petroleum Engineers has promoted the "black-oil simulation model," which is thought to be adequate for simulating 80 percent of the world's oil reserves. In this case, the application itself is the benchmark, as opposed to a specific code or algorithm.

One potential approach to computer performance measurement is to completely code an application—one that is representative and well defined—in a form suitable for optimally running on a given system. Unfortunately, this process must be repeated for each computer system earmarked for examination. This approach is fraught with difficulties stemming from the varying skills of human coders and the availability of appropriate system-level software.²

Every approach to benchmarking provides data concerning a computer system. The way this data is correlated with the user's anticipated workloads is almost never easy to understand. In addition, the reported literature offers practically no help in fostering an understanding of previous approaches

that have been found useful; this is especially true with previous failures.

If a proposed application (even if viewed as a specific algorithm coded in a specific language) could be decomposed into a set of kernel benchmarks, and each could be viewed as an impulse function in a multidimensional space, performance knowledge of existing benchmarks would allow performance prediction on the application as a whole.

One step in understanding whether such an approach can be realized is to provide an exhaustive collection of all benchmarks and benchmarking approaches as well as the results of those benchmarks and benchmarking approaches on all architectures in a form that the entire community can access, critique, and update.^{3,4,5}

The NBS Benchmark Collection. In June 1985, the NBS held a workshop in Gaithersburg, Maryland, to discuss techniques for measuring and evaluating parallel computers. At the conclusion of the workshop, a committee was formed to support, promote, and critique the NBS approach to a benchmarking collection.

The ad-hoc committee consisted of Ann Hayes of Los Alamos National Laboratory, Bob Keller of Quintas, Joanne Martin of IBM, John Riganati of the Supercomputing Research Center, and Leonard Uhr of the University of Wisconsin at Madison.

The collection, which is available online, includes benchmarks of three types: kernels, synthetic programs, and natural codes. NBS has attempted to define a common format for describing the benchmarks and tabulating the results of their use.

The concept calls for NBS to monitor and support the existence and evolution of the benchmark collection, and the community at large to contribute and describe benchmark usage.

NBS's original intent was to itself characterize the benchmark collection from time to time to promote its use and maximize its utility. Initially, the collection was weighted towards scientific computing and the majority of its elements were in Fortran. The collection recently added a number of programs for nonnumeric computation in the C language.

As we see it, the collection serves not only as a repository for benchmarks but also as a database of results giving anyone interested in performance measurement and characterization a broad base of experience from which he or she can draw.

One purpose of this article by two members of the committee is to cajole

the community to evolve this benchmarking collection more rapidly and vociferously. We suggest that editors of journals who regularly publish performance measurement and characterization studies alert their referees that the NBS collection exists and request that published results be routinely sent to the NBS for inclusion in the collection.

Programs established. A recent study⁶ by the National Research Council of the National Academy of Sciences strongly recommends development of a more substantive basis for performance

evaluation. The US Department of Energy and the National Science Foundation have responded to this call by establishing programs specifically focused on performance measurement and characterization.

The NBS benchmarking collection should evolve into a central part of these programs. To the extent their resources and mission requirements permit, other national laboratories should participate in this collection process—perhaps by maintaining specialty subsets in the areas of their particular interest and expertise.

Acknowledgments

We acknowledge the efforts of Ken Dymond, who played a key role in initiating the NBS benchmark collection, plus other members of the committee for their encouragement and constructive critiques.

References

1. "Performance Measurement," IEEE Subcommittee on Scientific Supercomputers, S. Fernbach, chair, Fall 1986, IEEE, Washington, DC.
2. J.P. Riganati and P.B. Schneck, "Supercomputing," *Computer*, Oct. 1984, pp. 97-113.
3. L. Uhr, "On Benchmarks: Dynamically

Improving Experimental Comparisons," Computer Science Tech. Report No. 571, University of Wisconsin at Madison, Dec. 1984.

4. S. Salazar and C. Smith, *Workshop on Performance Evaluation of Parallel Computers*, Nat'l Bureau of Standards, NBSIR-86-3395, July 1986.
5. J.L. Martin and D. Mueller-Richards, "Supercomputer Performance Evaluation: Status and Directions," *J. Supercomputing*, Vol. 1, No. 1, 1987, pp. 87-104.
6. "An Agenda for Improved Evaluation of Supercomputing Performance," Nat'l Research Council, Energy Engineering Board, National Academy of Sciences, Washington, DC, 1986.

NBS response

Shukri Wakid, National Bureau of Standards

The National Bureau of Standards' benchmark collection described in the accompanying article supports NBS research in measuring the performance of high-performance computers. The library of benchmark programs is available for evaluating processor performance and provides challenging computation problems that can be used to evaluate the efficiency of a system's design and software.

NBS invites organizations to use the programs that are currently in the system and to contribute other benchmark programs. However, the NBS regrets that it does not have the resources to evaluate and publish the data collected in benchmark studies. Other organizations may want to take on this activity.

The NBS collection is accessible via ARPAnet. It is configured as a three-level tree, where the first two levels are directories and the third level (leaf

nodes) contains the various files.

To access the library, send a netmail message to: nbslib@icmr.icst.nbs.gov. A reply message will be automatically generated with the desired contents.

The message should contain one of the following:

(1) "send index"—In response, you will receive the first-level directory containing the names of the second-level directories, plus a one-line description of their contents.

(2) "send index from <directory>"—In response, you will receive the contents of the specified <directory> similar in form to (1) above.

(3) "send <file> from <directory>"—In response, you will receive the contents of the specified <file> from the second-level <directory> indicated.

Participants sought to work on imaging interface task group

A task group called X3H3.8, Application Programming Interface (API) for Imaging, has been organized, according to X3, the Accredited Standards Committee for Information Processing Systems.

The new task group has been established to develop an American National Standard (ANS) on imaging by defining data objects, primitive operations, and reference models. The API developed will provide a means for building applications requiring imaging functionality within conventional, distributed, and imaging-oriented computing environ-

ments. The applications may include systems for medical imaging, machine vision, satellite and spacecraft imaging, and publishing.

Imaging processing, manipulation, and analysis are widely performed operations, but no current standard API exists in the imaging community and, until now, there has been no focused effort to develop a standard interface.

The lack of such a standard makes it difficult to produce and maintain applications that require imaging. It is becoming increasingly difficult to do this in contemporary environments that

use specialized imaging hardware or distributed and nonhomogeneous modeling of computing.

X3H3.8 expects to complete the draft ANS by January 1990, and interested persons and users are urged to get involved in the effort as soon as possible.

To join the task group, contact Patrick Krolak or Frank Drake, Center for Productivity Enhancement, University of Lowell, 1 University Ave., Lowell, MA 01854, phone (508) 452-5000, ext. 2693.

Second public review, comment under way on Extended Pascal draft standard

The second public review and comment period on draft proposed American National Standard X3.160-198x, Programming Language Extended Pascal, is in progress. According to an announcement by X3, the Accredited Standards Committee on Information Processing Systems, the public review period began in August and will continue through October 11.

Extended Pascal is an upward-compatible extension to existing Pascal standards and has been developed in an international effort between the American Joint X3/IEEE Committee, the British Pascal committee, the International Working Group on Pascal, and the ISO/JTC1 Subcommittee 22.

As such, Extended Pascal is being put forth as an American, British, and international standard designed to coexist with present Pascal standards until the existing ones are no longer needed.

Extended Pascal, a contemporary version of Pascal, the classic language for teaching programming concepts and programming languages, contains new features that reflect the modern software engineering techniques of modularity, protectability, and abstraction. In addition, Extended Pascal is a vibrant language containing many of the features that have been added to existing Pascal implementations.

The major features of Extended Pascal that distinguish it from the classic language defined by Jensen and Wirth are: modules with separable definition and interface parts (including initialization and finalization of modules), a generalized type definition mechanism (schemata) that permits static and dynamic selection from a "set of types," a secure method for initializing

variables of a given type (even if they are outside the module where the type is defined), and the ability to define private variables that have restrictions placed on their access.

In addition, Extended Pascal contains many pragmatic features that make it a more practical and usable language than classic Pascal, including: string types of both dynamic and fixed-length short-circuit logic operators, exponentiation, direct-access files, external file binding, complex numbers, constant expressions, record and array

value constructors, nondecimal integer constants, and time and date functions.

Extended Pascal also includes many simple conveniences that have become standard practice, such as an underscore character in identifiers, relaxed order of declarations, and an otherwise clause in case statement and variant records.

Copies of the draft standard may be obtained by calling Global Engineering Documents at (800) 854-7179. The single-copy price is \$35, and the international-copy price is \$45.50.

Software engineering subcommittee to meet

The fall meeting of the Computer Society's Software Engineering Standards Subcommittee will be held at 9 a.m. November 16 in Orlando, Florida, in conjunction with the Supercomputing 88 conference.

Since space for visitors will be limited, persons planning to attend are asked to notify the subcommittee's chair, John W. Horch, 705 Larry Drive, Madison, AL 35758-1116, phone (205) 532-1100.

Correction

The following projects have been completed and their proposed standards approved by the IEEE Standards Board:

- P959, I/O Extension Bus
- P982.1, Dictionary of Measures to Produce Reliable Software
- P982.2, Guide for the Use of Dictionary of Measures to Produce

Reliable Software

- P1028, Software Reviews and Audits
- P1096, Multiplexed High-Performance Bus Structure

The first paragraph above was inadvertently omitted from an article that appeared on p. 66 of the August edition.



CALL FOR PAPERS

NETWORK MANAGEMENT AND CONTROL WORKSHOP

September 19-21, 1989
Westchester Marriott Hotel
Tarrytown, New York

Technical Program Committee

Prof. Basil Maglaris
Polytechnic University
Dr. Salah E. Aidaros
Bell-Northern Research
Prof. Robert Boorstyn
Polytechnic University
Dr. Paul Callahan
AT&T Bell Laboratories
Prof. Wushow Chou
N. Carolina State Univ.
Dr. Roberta S. Cohen
AT&T Bell Laboratories
Prof. A. Ephremides
Univ. of Maryland
Mr. Leonard Fehskens
Digital Equipment Corp.
Mr. Dror S. Segal
Securities Industries
Automation Corporation
Dr. Jeffrey Jaffe
IBM
Dr. Gerard Louit
NYNEX Corporation
Dr. Patrick McGregor
Contel Corporation
Mr. Melvin Moskowitz
Merrill Lynch, Pierce
Fenner & Smith Inc.
Dr. Don Tow, Bellcore
Ms. Jil Westcott
BBN Labs, USA
Prof. Yechiam Yemini
Columbia University
Organizing Committee
Chairman
Prof. Ivan Frisch
Polytechnic University
Promotion & Publicity
Carol Lof
IEEE Communications
Technical Program
Prof. Basil Maglaris
Polytechnic University
CNOM Representative
Dr. Manu Malek
AT&T Bell Laboratories
Local Arrangements
E. Hart Rasmussen
Polytechnic University
Treasurer
Mr. Mark Wall
NYNEX Corporation

Authors are invited to submit unpublished contributions for a workshop on Network Management and Control. The workshop will be limited to 150 attendees to foster a personal interactive atmosphere in which there can be a creative discussion among working professionals, both developers and users. There will be keynote addresses, invited papers, theme papers, short papers, panel discussions, working sessions and wrap up sessions.

The focus of the workshop is real time network monitoring and control, with emphasis on the control of information systems, including both computers and communications.

TOPICS OF INTEREST IN NETWORK MANAGEMENT AND CONTROL

Control of multivendor and
multidomain systems

Artificial intelligence

Dynamic control

Protocols and standards

Measurement systems

User interfaces/
presentation systems

Software systems

Real time fault isolation

Integrated voice and data

Control of T1 and ISDN networks

Other related topics

AUTHORS SCHEDULE AND INSTRUCTIONS

February 1, 1989 — Submit five copies
each of cover page, abstract, and draft paper.
(Cover page should include paper title,
authors full name and affiliation, complete
address and telephone number).

April 1, 1989 — Notification of Acceptance
mailed to authors.

July 1, 1989 — Final papers due (5-10
pages).

Submit to: Professor Basil Maglaris
Technical Program Chairman
Center for Advanced
Technology in Telecommuni-
cations
Polytechnic University
333 Jay Street
Brooklyn, New York 11201

For further information contact Prof. Ivan
Frisch at the same address or call at (718)
260-3050.

Sponsored by Polytechnic University, Center For Advanced Technology in Telecommunica-
tions; NYNEX Corp.; and the Committee on Network Operations and Management
(CNOM) of the IEEE Communications Society.

Computer pioneer McCarthy wins Kyoto Prize

John McCarthy, who pioneered artificial intelligence over 30 years ago and won a 1985 Computer Society Computer Pioneer Award, has been selected as one of three winners of the 1988 Kyoto Prize.

McCarthy, a professor at Stanford University, along with Noam Abram Chomsky and Paul Thieme, will receive their Kyoto Prizes in November. Each prize carries a certificate of recognition, a medal, and 45-million yen (about \$344,000 in US dollars).

AI, Lisp work cited. The sponsoring Inamori Foundation of Kyoto, Japan, cited McCarthy for his work on AI and Lisp and will present him with its Kyoto Prize in Advanced Technology. The Computer Pioneer Award McCarthy won in 1985 was for the development of Lisp.

Director of the Artificial Intelligence Laboratory at Stanford, McCarthy is well known for his reasoning program approach to AI as pioneered in his 1958 paper, "Programs with Common Sense," published in the *Proceedings of the Symposium on the Mechanisation of Thought Processes*. The approach involves representing facts about the world—and the problem the program is to solve—as sentences of mathematical logic. The program must then use directed search procedures to deduce that a certain course of action will achieve the goal.

McCarthy had coined the term artificial intelligence and organized the Dartmouth Summer Research Project on Artificial Intelligence in 1956. This event united most of the individuals who were thinking about how to make computers behave intelligently.

In fact, he can be considered the father of artificial intelligence, a subject he has been relentlessly studying since 1949 after he attended the Hixon Symposium on Cerebral Mechanisms in Behavior at Caltech and began pondering the problem of making machines think.

With Claud Shannon, McCarthy coedited the 1956 Princeton University Press monograph, *Automata Studies*, which also contained the article he authored entitled "The Inversion of Functions Defined by Turing Machines" that formally defined the problem. The monograph stimulated further efforts in AI.

Inspired by the discussion, he and his students at the Massachusetts Institute of Technology defined Lisp in 1958. Today, this list processing language is still the most widely used language for AI studies.

In the 1960s, McCarthy developed the situation calculus formalism for representing the effects of actions. Much of the work in making AI programs that plan sequences of actions to



John McCarthy of Stanford University will be one of three persons to receive the Kyoto Prize in November.

achieve goals is based on situation calculus or its variants.

In the 1970s, he developed the circumscription method of nonmonotonic reasoning programs described in his April 1980 paper on artificial intelligence.

Two other winners. Also at the November ceremonies, Chomsky will receive the Kyoto Prize in Basic Science for his work in linguistics, specifically his contribution to generative grammar theory, and Thieme will receive the Kyoto Prize in Creative Arts and Moral Sciences for his philological studies of Vedic literatures.

The Inamori Foundation was established in 1984 with a 20-billion yen endowment from Kazuo Inamori, founder and chairman of the Kyocera Corporation. Since 1985, the principal activity of the foundation has been the presentation of three annual Kyoto Prizes.

The foundation awards the Kyoto Prizes to recognize individuals or groups "who have made outstanding contributions to the development of science and civilization or otherwise to uplifting and betterment of human spirit and morale."

ABET program evaluators sought

The Computer Society is seeking qualified program evaluators for accreditation of computer engineering programs by the Accreditation Board for Engineering and Technology (ABET).

The ABET Accreditation Committee of the society's Educational Activities Board is conducting the search for nominees to be considered for recommendation as evaluators.

For consideration, a candidate must hold an MS or PhD degree, must have

at least 10 years' professional experience in computer engineering, must be a fellow or senior member of the IEEE, and must have present or previous faculty involvement at an ABET-accredited institution.

Persons interested in nominating themselves or colleagues may obtain complete information by contacting Fred Maryanski, Dept. of Computer Science and Engineering, U-155, University of Connecticut, Storrs, CT 06268, phone (203) 486-2584.

Robotics TC active in conferences, transactions

Mohan M. Trivedi, Chair, Robotics Technical Committee

With membership in the Computer Society's Robotics Technical Committee experiencing steady growth and reaching a current figure of about 1,000, the TC has taken a number of initiatives in 1988. These include active involvement in conferences and transactions.

Last April, the Robotics TC and the society cooperated with the IEEE Systems, Man, and Cybernetics Society and the International Society for Optical Engineering in organizing the annual Applications of Artificial Intelligence Conference in Orlando, Florida. The three-day event had about 700 attendees, making it one of the largest of its kind in the country.

The three plenary sessions each featured a keynote address, one by B. Chandrasekaran of Ohio State University discussing computational aspects of intelligence, another by Azriel Rosenfeld of the University of Maryland presenting an overview of computer vision, and a third by Robert Simpson of DARPA discussing AI research and progress from the Department of Defense perspective.

The event also included 21 technical sessions and 86 paper presentations. Those dealing with robotic systems, path planning, architectures for AI, knowledge-based systems, computer vision, and knowledge representation were of particular interest to robotics-area professionals.

The next conference will also be held in Orlando, in the latter part of March 1989. Complete information regarding the call for papers, tutorials, and special sessions may be obtained by writing to Mohan Trivedi, Electrical and Computer Engineering Department, University of Tennessee, Knoxville, TN 37996-2100; by telephoning (615) 974-5450; or by E-mail at trivedi@vms1.engr.utk.edu.

The Robotics TC participated in the three-day Computer Society-sponsored Symbiotic and Intelligent Robotics Workshop at the University of Maryland August 29-31. The theme dealt with investigation of the languages and the automation aspects in the development of intelligent robotics systems and their symbiosis with humans.

The TC is one of the eight Computer Society technical committees collaborating to stage Compusat 88 (see story in adjoining columns). The satellite symposium will be aired October 4 at several hundred sites all over North

America, with some 10,000 participants from universities, industry, and local IEEE chapters viewing the live two-way proceedings.

Specifically, the Robotics TC will present a minitutorial, research highlights, demonstrations of state-of-the-art applications, a panel discussion, and a question-and-answer session.

In the publishing field, the Robotics TC is cooperating with the IEEE Sys-

tems, Man, and Cybernetics Society in developing two special issues of the society's transactions. The first issue, devoted to computer vision, will be published in July 1989, and the second, dealing with unmanned autonomous systems, will be published in December 1989.

Both issues should be of special interest to professionals in the robotics field.

Bill introduced to repeal Section 1706 of Tax Code

The Computer Society's Committee on Public Policy has learned that Senator Daniel Patrick Moynihan (D-NY) introduced legislation July 11 to replace Section 1706 of the 1986 Tax Reform Act.

According to the July 25, 1988, issue of the *IEEE USAB Legislative Report*, the new bill, S.2626, defines a new statutory "safe harbor" test for determining employment status of individuals providing technical services as engineers, designers, drafters, computer programmers, systems analysts, or similarly skilled workers.

S.2626 was referred to the Senate Finance Committee. Judging from the financial impact statement, raising the estimated revenue from \$12 million per year to \$100 million, the new provisions apparently no longer single out the computer profession, but cover other professions as well.

COPP plans to keep the society membership posted on development of the new legislation, which "is to take effect on the date of enactment." The current draft apparently contains no provisions for transition relief.

Persons enlisting Compusat sites can earn awards

Compusat 88 is conducting a contest, with awards going to those enlisting certain numbers of sites for the Computer Society's inaugural satellite symposium, slated Tuesday, October 4 from 11:30 a.m. to 4:30 p.m. Eastern Time.

Each person enlisting three sites for Compusat will get a Founder Award, and anyone lining up six sites will receive a Pioneer Award. The individual enlisting the most total sites will win the Compusat Person-of-the-Year Award.

To submit your list of sites, or to obtain more information, contact Betty Maranca, IEEE Service Center, PO Box 1331, Piscataway, NJ 08855-1331, phone (201) 562-5494.

Compusat is being cosponsored by the CS Technical Activities Board and eight Technical Committees under the theme "The Interdisciplinary World of Computing." The live videoconference will use one-way video and two-way audio.

It will feature talks by prominent computer-industry figures, and current-technology presentations by the eight TCs, plus companies, universities, and governmental agencies on diverse subjects from advanced tools and techniques to very large scale integration.

The schedule will be as follows (for the Eastern Time zone): 11:30 a.m., introduction; 11:38 a.m., very large scale integration; 12:03 p.m., design automation; 12:28 p.m., test technology; 12:58 p.m., questions and answers (nationwide audience); 1:15 p.m., lunch; 1:45 p.m., software engineering; 2:10 p.m., personal computing and office automation; 2:40 p.m., Q and A; 2:55 p.m., machine intelligence and vision; 3:20 p.m., robotics; 3:50 p.m., Q and A; 4 p.m., future directions and challenges; 4:10 p.m., Q and A (entire panel); 4:25 p.m., conclusions.

Additional details may be obtained by referring to the article on p. 77 of the August issue of *Computer*.

NEWS FROM THE COMMITTEE ON PUBLIC POLICY

Copyright Office rules display and program to have single copyright

Ralph J. Preiss, COPP chair

After considering the matter for nine months, the US Copyright Office has ruled that it will henceforth require screen displays and the computer programs with which they are associated to be registered as a single work. This will be the case whenever the two elements are owned by a single person or business entity.

The Copyright Office published its decision in a June 3 "Notice of Registration Decision."

"Permitting multiple registrations of parts of works would increase confusion among those attempting to use the records of the Copyright Office," the Copyright Office argued.

"Subdividing claims might also result in multiple infringement actions and multiple claims for statutory damages, based on separate registration. The Copyright Office benefits by having a simplified administrative process," the notice said.

The Copyright Office further stated that it "is sympathetic to users who may have difficulty in determining the scope of the copyright in computer software," but that's a problem that the Copyright Office cannot deal with and one the courts will have to sort it out. The most the office feels it can do to assist the public and the courts is "to refuse to register any clearly frivolous, unsound, or otherwise unjustified claims."

Essentially, this leaves the interpretation of copyright violations of computer screen displays up to litigation and opens the way for expensive—lucrative to some experts—court battles.

Input was invited. The Copyright Office invited technical input on proposed rulemaking for copyright registration of screen displays on July 16, 1987. In response to this invitation, the Computer Society, acting through its Committee on Public Policy, prepared and submitted a position paper on whether screen displays should be considered part of the underlying computer programs with which they are associated or separate works of authorship.

Comments were also offered on the likely consequences for software economics and software progress if screen displays and the underlying computer programs were to be registered

The ballot box

Should the Computer Society appeal to the Copyright Office on its June 3 decision? Please respond by circling 180 on the reader response card if you want COPP to appeal the decision; or circle 181 if COPP should accept the decision and not file an appeal.

separately or together for copyright purposes.

The society's Board of Governors approved the following motion via electronic mail ballot tallied September 8, 1987:

RESOLVED, that the Board of Governors of the Computer Society of the IEEE, in order to encourage and promote creativity and investment in this aspect of the computer graphics field, recommends that the Copyright Office of the Library of Congress should permit an owner of a computer program screen display to register the screen for copyright protection apart from the registration of the underlying code in the computer program that generates the display.

The board also approved presentation of the written comments on this issue as submitted to the Copyright Office by Richard H. Stern, chair of the Subcommittee on Intellectual Property of the society's Committee on Public Policy.

Comments were sought. The next day, COPP representatives, including Stern, orally explained the position paper at a Copyright Office hearing on the issue. Comments were specifically invited in the following areas:

(1) Should screen displays be registered separately from the underlying computer program that generates it under any circumstances? The present situation combines programs and screens in one copyright.

(2) If the response is yes, specify the circumstances under which separate registration should be made.

The society's position paper asserted that a menu or other screen display should be registrable separately from the underlying program if the display is

original and otherwise satisfies the statute.

(3) Should a distinction be made between the registration of a textual and a pictorial display?

The society's position paper asserted that there appeared to be no sound reason for distinguishing between textual and pictorial displays.

(4) What should be the appropriate deposit if any registration is made, either as part of a computer program registration or separately?

The society's position paper asserted that printouts and photographs of screens would be the most appropriate deposits or identifying material (to the Copyright Office).

(5) What is the relationship between a computer program and the generation of a screen display? Describe the technology and methods of creating displays.

(6) Can completely different computer programs produce a substantially similar screen display? Explain the technology. Discuss the degree of similarity.

The society's position paper provided a description of the technology involved in creating displays and established the point that many different computer programs (and languages from which the programs are generated) can create the same or similar voltages/currents that are used to reproduce a display, that the intellectual content of a screen—its "user-friendliness"—is really the key to its usefulness, and this has nothing to do with the code that generated it.

A further set of discussions delved into the three types of copyright protection presently offered: VA, for pictorial or graphic work (examples are maps, which can be like computer-generated instructions or menus); PA, for audiovisual work, such as video games; and TX, for programs as literary work. Again, the case was made that the display and the driver are separate and should therefore be granted separate copyrights.

(7) How will our registration decisions affect the public interest in encouraging the development of new and improved computer programs? Will this interest be helped or hurt?

The society's position paper asserted that separate registration would encourage innovation because software designers would be encouraged to avoid infringement of proprietary material.

Society's board backs report urging upgrading computer science schooling

The Computer Society Board of Governors has unanimously endorsed a report recommending "excellence in undergraduate computer science education ... be made a matter of high priority for our nation."

The board endorsed the Undergraduate Computer Science Education Report June 17, a day after the society's Educational Activities Board and Executive Committee gave the report their unanimous approvals.

The report evolved from a March 1988 workshop at George Washington University sponsored by the National Science Foundation under NSF Grant CCR-8811841.

Executive summary. According to Gerald L. Engel, education vice president of the society, the workshop participants issued an "executive summary" of the report. The summary states:

"Computer science is a relatively new and dynamic field. In just over 40 years, it has become a central technological influence in our society and a key element in our continuing economic development, international competitiveness, and national security.

"To use computers effectively, the nation must have a continuing supply of well-educated computer professionals. Further, an understanding of the central ideas concerning computers is essential to all college graduates, who will increasingly use computers. Therefore, excellence in undergraduate computer science education must be a matter of high priority for our nation.

"Computer science and the computer industry have experienced explosive growth. There have been staggering advances in the depth and breadth of our knowledge of computer science and nearly exponential growth in the number of computer science degree programs. Computers have improved in capacity and performance and fallen in price at an extraordinary pace.

"New computer science knowledge has necessarily led both to a need for continuing revision of curricula and instructional materials, and to a need for retaining and revitalizing many current faculty. An expanding number of undergraduate degree programs and insufficient rewards for undergraduate educational activities have caused serious shortages of qualified faculty, especially in non-PhD-granting departments.

"Although computer science PhD production has recently improved, it

still falls far short of satisfying the nation's demands. For example, there are over 1,000 computer science degree programs in the nation's colleges and universities. Most of these programs are in four-year institutions, do not offer the PhD, and have no or only a few faculty who themselves hold a computer science PhD.

"Yet, in 1986-87, only 41 of the 466 new computer science PhDs took teaching positions in the non-PhD granting computer science degree programs. This contrasts, for example, with the 202 of 845 new mathematics PhDs who took teaching positions in the non-PhD granting mathematics degree programs.

"Computer science is a laboratory science, yet many undergraduate computer science departments have not yet been able to establish instructional labs. Even where labs are in place, the advent of powerful workstation computers means that many existing labs need to be modernized. Instructional materials need to be created to take advantage of new opportunities for better instruction. Opportunities for using new media

or improved instructional strategies to improve instructional delivery in a wider educational context should be exploited."

Areas of concern. "To identify solutions to these concerns, this NSF-sponsored workshop brought together 32 computer scientists who organized their efforts by focusing attention on problems in four separate areas of concern: curricula, faculty, laboratory infrastructure, and instructional delivery. Solutions which were both highly ranked and for which there was a broadly based consensus are listed below; order does not imply priority.

"In the area of curricular development, we recommend that the NSF establish two national centers to serve as focal points for computer science curricular development, training, and dissemination. These centers will focus the energy and attention required to create and maintain the curricula, instructional materials, and training opportunities needed to bring undergraduate computer science instruction

THE SECOND INTERNATIONAL

Software for Strategic Systems Conference

Von Braun Civic Center ▲ Huntsville, Alabama ▲ October 25-26, 1988

This conference will provide the unique opportunity for participation in an international forum emphasizing software issues of strategic importance. Included in the program are speakers on a national and international level who can speak authoritatively about strategic issues. Prominent technologists will also discuss recent technology advances within this critical software development area. Keynote speakers will include: Lt. Gen. James A. Abrahamson, Strategic Defense Initiative Organization; Dr. Edith W. Martin, Boeing Electronics Company; and Mr. Charles A. Zraket, MITRE Corporation.

For further information or a brochure, call (205) 895 - 6372.



The University
Of Alabama
In Huntsville

Division of Continuing Education
Conferences
Tom Beville Center
Huntsville, Alabama 35899

An Affirmative Action/Equal Opportunity Institution

up to date and to maintain it in the face of continuing rapid change.

"To focus more national and faculty attention on teaching, we recommend that the NSF implement presidential young teacher awards, which will provide incentives and rewards for creative and successful teaching of undergraduates, indicate to administrators, faculty, and students that both teaching and research are significant, and bring national attention to the importance of educational excellence.

"To provide adequate laboratory infrastructure, the NSF should significantly expand the Instrumentation and Laboratory Improvement Program to create and update effective laboratory

infrastructures in a substantial fraction of the nation's undergraduate computer science degree programs.

"To improve instructional delivery, the NSF should support research into new instructional technologies to realize the opportunities that computer science has to lead the academic community in general enhancement of instruction through computer technology.

"The workshop participants are pleased that NSF has recently begun to respond to the need for increased emphasis on undergraduate education. Some of our proposals call for expanding existing programs; others, for fundamentally new programs.

"The importance of computer science

undergraduate education to the nation dictates that additional funding be found to support the initiatives proposed in this report. Our recommendations address the creation and updating of curricula, the enhancement of faculty knowledge, development and improvement of laboratory infrastructure, creation of instructional materials, and use of new computing equipment to improve instructional delivery."

Reports available. Full copies of the report may be obtained by contacting James Foley, Department of Electrical Engineering and Computer Science, George Washington University, Washington, DC 20052.

Moore, Noyce co-recipients of Computer Entrepreneur Award

Intel Corporation Chairman Gordon E. Moore and Vice Chairman Robert N. Noyce have been presented the Computer Society's 1986 Computer Entrepreneur Award.

Rex Rice, representing the society's Awards Committee, made the presentations in Noyce's office in Santa Clara, California, June 7. Moore was unable to attend the ceremony due to a travel-schedule delay. The presentations were originally scheduled for Compcon Spring 1987, but had to be postponed several times because of the awardees' heavy travel commitments.

Moore and Noyce were selected for "entrepreneurial contributions to the semiconductor memory and microprocessor industries," as stated in the resolution of the society's Board of Governors. They have been responsible for making Intel a microprocessor-on-a-chip giant.

The two men previously won the society's Computer Pioneer Award for "integrated circuit production technology." Moore also received the 1978 W. Wallace McDowell Award for "outstanding contributions to research and development of semiconductor components and his insights and leadership in the microprocessor and semiconductor memory fields."

Rice chaired the Computer Entrepreneur Award subcommittee of the CS Awards Committee in 1986 when Moore and Noyce—among others—were nominated for Board of Governors ratification.

The board established the Computer Entrepreneur Award in 1985 to recognize and honor technical managers whose outstanding leadership fostered



Robert N. Noyce (left) accepts the 1986 Computer Entrepreneur Award from Rex Rice, who chaired the subcommittee that selected Noyce and Gordon Moore (not pictured) in 1986. Noyce displays his newly won silver cup, and Rice holds the cup presented to Moore in absentia.

the growth of a "segment of the computer industry." It is a companion to the Computer Pioneer Award established in 1982, the society's 30th anniversary.

The Entrepreneur Award is a sterling silver champagne cup commissioned by the society to Kurt Matzdorf, a distin-

guished New York silversmith. The gold-plated crown below the cup displays the alternating IEEE and Computer Society symbols supported by laurel-leaf clusters, the ancient symbol of outstanding achievement. The recipients names are engraved around the cup's base.



Computer Society Election

Nominees for Computer Society office and Board of Governors

On the following pages are the position statements and biographies of the Computer Society's candidates for president-elect, first and second vice presidents, and Board of Governors. Within each category, candidates are listed in alphabetical order.

Election of officers to one-year terms and of board members to three-year or two-year terms, each beginning January 1, 1989, will be by vote of the membership as specified in the bylaws. Ballots, which will be mailed to all society members about September 2, must be at IEEE Headquarters no later than October 14, 1988. Election results will be announced in the December issue of *Computer*.

The opinions expressed in the statements are those of the individual candidates and do not necessarily reflect Computer Society positions or policies.

Nominee for president-elect



Helen M. Wood

Position statement.

In recent years, the Computer Society has experienced rapid growth in membership and expansion of services, followed by a ma-

jor drop in income and belt-tightening as a result of the computer industry slump. Now, membership is again growing and conference attendance and publication subscriptions are on the increase. What has been learned from this experience?

While we have always recognized technical quality as an important attribute for Computer Society activities, recent experience has taught us some additional, important lessons. Among them is that growth must be well supported and carefully managed, with serious attention to financial reserves and volunteer and staff workload. Where we are applying these principles, we are realizing positive results.

A case in point is our standards program, which I have had the opportunity to oversee for the past two and a half years. During that time, this activity has grown from 45 to

over 150 standards and projects involving over 5,000 participants. These standards have widespread effects on products and practices throughout all sectors of the global economy, so it is important that their development proceed in a well-organized manner, without undue delay or expense.

Last year, the Board of Governors approved the hiring of a full-time standards support coordinator and a significant increase in budget. Now, we are seeing improvements in the support and publicity for our standards committees. In addition, we are identifying alternative sources of financial and administrative support for a number of committees. This will allow our standards program to continue to expand without danger of becoming a financial burden.

Although much progress has been made in a number of areas, opportunities remain for additional improvements in society services and operations. The need for improved vitality in the society's technical committees has been widely recognized for some time. The potential of the Computer Society Press has not been fully realized.

Drawing upon my past experience with the society and in the management of technical organizations, I will work to identify and implement needed improvements to keep the society healthy and strong. I welcome your support and ideas.

Biography. Wood is currently the Computer Society's first vice president. She has been

vice president for standards since 1986, and served on society and IEEE boards and committees, including the Board of Governors, Standards Activities Board, Technical Activities Board, Finance, Executive, and Operations Committees, and the IEEE Standards Board and Budget Development Committee. Wood was formerly Computer Society treasurer and is currently editor of "Standards" for *Computer*.

Wood was recently appointed director of the Office of Satellite Data Processing and Distribution for the National Oceanic and Atmospheric Administration (NOAA), National Environmental Satellite, Data and Information Service. Prior to this appointment, Wood was the deputy director of the Institute for Computer Sciences and Technology at the National Bureau of Standards. In previous positions at NBS, she directed the Center for Programming Science and Technology, Center for Computer Systems Engineering, and Information Systems Engineering Division. Earlier she performed research in computer networking, computer security, and process control.

She holds the BS in mathematics from the University of Maryland and the MS in computer science from American University. She is a senior member of IEEE, a member of ACM and the Association for Women in Computing, and a recipient of Department of Commerce Bronze and Silver Medals and several Computer Society awards.

Nominees for first vice president



Barry W. Johnson

Position statement.

The Computer Society's purposes are to advance the theory, practice, and application of computer and information processing tech-

nology and to promote the exchange of technical information among its members. Fulfilling these purposes requires (1)

excellent sources of technical information and innovation, (2) varied and high-quality membership services, (3) financial stability, and (4) visibility of our activities.

The society's technical committees are excellent sources of technical information. I will work with volunteer leaders, TC members, and staff to enhance TC activities and visibility and to increase TC participation by our members.

We must continue to seek new and better member services such as publications, conferences, and standards. Perhaps more important, however, is assuring the technical quality of our existing programs. I will work

diligently to ensure the highest quality in all of our member services and to solicit input from our members concerning quality.

The Computer Society must be fiscally sound in order to continue its activities and maintain healthy growth. I will work for long-range budget planning that anticipates the cycles that commonly occur in the computer industry. We must plan for financial cycles so that we can maintain a consistent level of funding for our important activities.

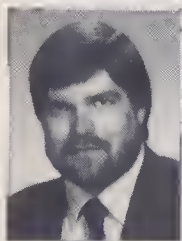
Finally, the activities of the Computer Society need international visibility, both among the membership and the profession in general. I will support enhanced promotional efforts for recruiting new society members, keeping existing members, and increasing general awareness of society activities.

Biography. Johnson, currently the Computer Society treasurer, has been active in the society for several years. He has chaired the Membership Development Committee and the Finance Committee and served on the Conferences and Tutorials Board, the Technical Activities Board, and the Membership and Information Board. An

associate editor of *IEEE Micro*, Johnson has served as guest editor for several special issues.

While with Harris Corporation, Johnson participated in fault-tolerant system design and analysis. Currently, he is an assistant professor of electrical engineering and a member of the Center for Semicustom Integrated Systems at the University of Virginia. His research interests include fault-tolerant computing, testing, and VLSI. He is the author of a new textbook, *The Design and Analysis of Fault-Tolerant Digital Systems* (Addison-Wesley).

Johnson received his PhD, ME, and BS in electrical engineering from the University of Virginia. He has received an award for outstanding contributions from Eta Kappa Nu's Virginia chapter, an outstanding young faculty member award from the Department of Electrical Engineering, and a Certificate of Appreciation and a Meritorious Service Award from the Computer Society. An IEEE member, Johnson also belongs to the Industrial Electronics Society, Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.



Joseph E. Urban

Position statement.

The Computer Society has returned to a good growth period. I believe this growth should be reflected in improved quality coupled with

increased quantity of member benefits. We must continue to control costs on publications and technical activities while maintaining high quality. I believe that the key to controlled growth is based on sound technical and financial plans.

My goals are as follows:

(1) expanded growth in conferences for new thrust areas in which there would not be duplication of effort;

(2) enhanced information flow to members on technical material and society benefit programs through existing publications and potential new sources;

(3) advanced visibility for standards working-group activities to promote the international aspects and stimulate further growth; and

(4) increased emphasis for grass-roots technical efforts in chapter and regional activities where the entrepreneurial spirit has resulted in well-defined plans.

I will continue to support non-US-based technical activities such as conferences, lectures by visiting speakers, networks, and society offices. I believe in improving the working-environment aspects of the Computer Society offices. These improvements are designed to ensure the outstanding level of staff support required by members.

I will work to achieve the goals outlined above to build on the solid foundation of the Computer Society.

Based on my past experience with the Executive Committee and the Area Activities, Conferences and Tutorials, Publications, and Technical Activities Boards, I believe that I can continue to serve the society in a positive manner.

Biography. Urban is the Computer Society's second vice president for conferences and tutorials, an editor of *IEEE Transactions on Software Engineering*, and general chair for the 1988 International Symposium on Databases in Parallel and Distributed Systems. He has served as a member of the

Board of Governors and Executive Committee, treasurer, and Finance Committee chair.

He initiated and chaired the Computer Languages Technical Committee and chaired the Publications Planning Committee. He chaired and lectured in the Chapter Tutorials and Distinguished Visitors Programs. He was conference chair for three conferences on Ada, logic programming, and computer languages.

Urban is professor of electrical and computer engineering and associate dean for research in the College of Engineering at the University of Miami. Previously, he worked at the US Army Signal Center, the University of South Carolina, Columbia, and the University of Southwestern Louisiana. His research areas are in software engineering and computer languages.

Urban earned a BS in computer science from the Florida Institute of Technology (1973), an MS in computer science from the University of Iowa (1975), and a PhD in computer science from the University of Southwestern Louisiana (1977). He received an ACM Doctoral Forum award in 1978.

Nominees for second vice president



Gerald L. Engel

Position statement.

The success of the Computer Society in the past few years has been phenomenal. In a real sense, the main thing that any candi-

date must pledge is not to disrupt this successful period of growth and expansion.

My interest focuses on the Computer Society expanding its leadership role in a variety of areas. I would like to see continuing efforts in our accreditation activities through both CSAB and ABET. I would like to see additional cooperative activities with ACM and other appropriate sister societies. I would like to see the Computer Society expand its influence within IEEE. I would like to see our activities and influence in the national science and engineering community expand. I would like to see our international role in the science and engineering community expand. All this seems even more urgent with the inactivity of AFIPS in the past few years.

Careful management and leadership must be extended to see that our areas of success,

such as publications, conferences, and standards, continue. From my previous experience in these kinds of activities, I believe I am uniquely qualified to assist in such an effort. Also, the increasing autonomy of the technical committees must continue to the point that they become the recognized leaders in all of their respective areas.

Finally, if elected, I will do all I can to provide an effective channel of communication between the membership and the Executive Committee and the Board of Governors. I have enjoyed my work in the Computer Society in the past, and look forward to the opportunity to provide further leadership and service. Thank you.

Biography. Engel is currently the Computer Society's vice president for educational affairs. He previously served as vice chair of the Educational Activities Board and in other capacities on that board and the Publications Board.

He has, throughout his professional career, been actively involved in various

aspects of computer education. Included in this has been active participation in the development of model curricula for both the Computer Society and ACM.

Engel currently serves as a representative director for the Computer Society on the Computing Sciences Accreditation Board and has just completed a term as chairperson of the Computer Science Accreditation Commission. He has also been involved in the development of several conferences dealing with computers in education. He was one of the founding members of the Steering Committee of the National Educational Computing Conference and served as chairperson of that body from 1979 to 1985. In 1985, he served as general chairperson of the IFIP World Conference on Computers in Education.

Engel occupies the Stamford Chair of Computer Science and Engineering at the University of Connecticut at Stamford. Prior to this position, he held teaching and research positions at Christopher Newport College, Old Dominion University, and the Virginia Institute of Marine Science. He holds the doctor of education degree in computer science from Pennsylvania State University.



Laurel V. Kaleda

Position statement.

The Computer Society's growth is the result of strong leadership in our field and a solid blending of theory and practice in our activi-

ties. The society's resulting size, complexity, and diversity require that its leaders work to ensure efficient, effective operations while supporting continued expansion and enhancement of membership services. This challenge is compounded by the rapid advancement of computer technology and the current economic climate.

To sustain our current growth and continue improvement of services, we need to focus on three specifics:

- (1) ensuring that our members are aware of the wide range of society-sponsored activities,
- (2) encouraging all of our members to become actively involved in one or more of these activities,
- (3) continuing the process of self-assessment to periodically review what's

good, bad, timely, or out-of-date and then focusing more effort on the good and timely and establishing plans to change or eliminate the bad or out-of-date.

As vice president for technical activities, I have focused the efforts of the Technical Activities Board to seek out new ways of getting more information to members about the society's technical activities. This is crucial to the society's continuing health, and it encourages members to be more active.

The previous and current Boards of Governors and officers have substantially improved society operations so that growth in services and membership can be supported effectively. They have managed this with strong fiscal responsibility.

If elected, I will work to continue this progress.

Biography. Kaleda is the Computer Society's vice president for technical activities and a member of the Board of Governors. She is active on the Membership and Information Activities Board and has served on the

Finance and Audit Committees. She chaired the society's Standards Coordinating Committee, was program cochair for Compstan 88, the Conference on Computer Standards, served on several program committees, and organized and chaired a number of panel sessions in the standards area. She was awarded a Meritorious Service Award in 1987 for her excellence in leadership and support of the society's standards activities.

Kaleda is a member of the Storage Systems Strategy Department for IBM's General Products Division, San Jose, California. IBM GPD develops and manufactures disk and tape peripherals, as well as major software products. Previously, she was division standards authority for IBM GPD. Her 20 years with IBM have provided experiences on a wide range of computers (from PCs to 3090s) and software (operating systems through applications).

Kaleda has a BS in applied mathematics and computer science from Washington University, St. Louis, and an MBA from Golden Gate University, San Francisco. She is a licensed professional engineer in California, a senior member of IEEE, and a member of ACM.

Nominees for Board of Governors (Vote for 14)

The seven highest vote-getters among the 23 candidates for Board of Governors will serve three-year terms; the seven next highest will serve two-year terms. This procedure begins the transition, specified in the fall 1987 constitutional amendment, from a 20-member board of two-year positions to a 21-member board of three-year positions with one-third of the members elected each year.

Computer Society Election



**Return
your ballot
promptly.
Ballots must be
received at
IEEE Headquarters
no later than
October 14, 1988.**



Tilak Agerwala

Position statement.

The Computer Society is recognized worldwide as a well-run organization that provides many important services to its membership. One of the

most important is to facilitate exchanging information through journals, magazines, conferences, workshops, and tutorials. As a member of the Board of Governors, I will actively support these forums and encourage

- continued focus on *Computer* magazine to ensure that it maintains its vitality,
- new workshops in emerging areas,
- timely and relevant tutorials,
- better technical committee newsletters, and
- increased use of computer networking.

Another important society role is to recognize the professional contributions of computer scientists and engineers. The society has an extensive awards program that can be strengthened. For example, compared to other IEEE societies, we have the lowest percentage of members who are IEEE fellows. I will work toward

- expanding the awards program,
- identifying qualified candidates and soliciting nominations, and
- increasing the number of qualified senior members and fellows.

Biography. Agerwala has served the Computer Society in several capacities. He is currently chairperson of the Computer Society Fellows Committee. He has served as a member of the Awards Committee, a distinguished visitor, chairperson of the Eckert-Mauchly Award Committee, chairperson and vice chairperson of the Technical Committee on Computer Architecture, and chairperson and vice chairperson of the Central Texas Chapter. He has been active in the International Symposium on Computer Architecture and was a co-program chairperson in 1985.

Agerwala is currently a member of the Corporate Technical Committee of IBM. He was director of symbolic and numeric processing at the IBM T.J. Watson Research Center and was responsible for research programs in artificial intelligence, engineering/scientific computing, and parallel processing. Before joining IBM in 1979, he was an assistant professor at the University of Texas at Austin.

Agerwala received his BTech in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1971 and his PhD in electrical engineering from Johns Hopkins University in 1975.

Agerwala is a fellow of the IEEE and a member of ACM. He received the Samuel N. Alexander Memorial Award from the Washington, DC, chapter of ACM and an Invention Award and Outstanding Technical Achievement Award from IBM.



Vishwani Agrawal

Position statement.

There are two aspects of the Computer Society that we should preserve and enhance. The first is its strong technical nature. Through its publi-

cations, conferences, workshops, and technical committees, the society provides a meeting ground for professionals of similar interests. This is the only forum where organizational boundaries melt away. The Computer Society serves the profession and the professional alike. I will strive to maintain the technical nature of the society.

Second, and equally important, is the international nature of the society. Advances in computers are influencing life around the globe. Society activities provide a neutral ground for individual professional growth and business interactions. Our liaison with other professional societies has been increasing. Recently, I was instrumental in initiating a dialogue with the Computer Society of India. I strongly believe that the Computer Society is destined to play a leading role internationally in the computer and communications revolution.

Biography. Agrawal served as editor-in-chief of *IEEE Design & Test of Computers* from 1985 through 1987. During that period, he also served on the Publications Board and the Magazine Advisory Committee. During 1984-85, he was the operations vice chair of the Computer Society's Design Automation Standards Subcommittee. He has been active in technical program committees of the Design Automation Conference, Fault-Tolerant Computing Symposium, International Test Conference, Built-In Self-Test and Design for Testability Workshops, and a Japanese AI/CAD workshop. He was a co-organizer of the First VLSI Design Workshop, held in India in 1985 and jointly sponsored by the IEEE and the Computer Society of India.

Agrawal is a member of technical staff at the AT&T Bell Laboratories, Murray Hill, New Jersey, where he has significantly contributed to CAD research and tool development for VLSI design verification and testing. He holds a PhD in electrical engineering from the University of Illinois at Urbana-Champaign. The author of 90 papers, he holds one patent and has received four best paper awards. He received the Outstanding Contribution Certificate from the Computer Society. He is a fellow of the IEEE, a fellow of the Institution of Electronics and Telecommunication Engineers (India), and a member of the ACM. He coauthored a recent Computer Society Press tutorial book titled *Test Generation for VLSI Chips*.



Mario R. Barbacci

Position statement. The Computer Society is a remarkable organization in the range and quality of services it provides to its members and to the profession.

It sponsors or cosponsors most of the field's major professional meetings; its magazines and transactions provide a forum for scholarly articles as well as surveys and tutorials; its Computer Society Press is one of the largest book publishers in the computer field.

But these services are only part of the picture. The society is also people—members who volunteer their time, energy, and intellect to advance the profession and, in turn, reap the benefits of technical growth and peer-network development. Through technical and standards committees, members participate in important technical activities and grow professionally. Unfortunately, the opportunities these activities provide are among our best kept secrets.

As a board member, I will work to increase the visibility of volunteer activities and the opportunities they present for professional growth.

Biography. Barbacci is a member of the Board of Governors, vice chair of the Technical Activities Board, a member of the Long-Range Planning Committee, a member and past chair of IFIP Working Group 10.2 (Computer Descriptions and Tools), and a category editor for *ACM Computing Reviews*.

He is a senior member of the Software Engineering Institute and a member of the computer science faculty at Carnegie Mellon University. A founder of SEI, he was responsible for the technical proposal that led to its establishment and served as its associate director for technology exploration from January 1985 to April 1986. He currently leads a project investigating languages and methodologies for distributed applications on heterogeneous computer networks.

Barbacci received the BSEE and engineer degrees from Universidad Nacional de Ingenieria, Lima, Peru, and the PhD in computer science from CMU. His doctoral research laid the foundation for the CMU Design Automation Project. Later, he designed and implemented the facility used in DoD architecture evaluation experiments leading to the Military Computer Family and was responsible for the design and implementation of Spice Ada.

He is a senior member of IEEE, a member of ACM and Sigma Xi, and a recipient of the IFIP Silver Core Award.



P. Bruce Berra

Position statement. My main goals are to increase the quality and quantity of services to the membership. If elected, the following are some of the items that I will emphasize.

(1) Strengthen *Computer*. Our flagship magazine is the best of its kind and should be significantly increased in size.

(2) Begin a Distinguished Tutorials Program. The Distinguished Visitors Program is extremely successful. I believe that we should extend the concept to tutorials.

(3) Increase student membership. Students are the lifeblood of our society. We need to find new ways of attracting them and making the transition from student to regular membership as painless as possible.

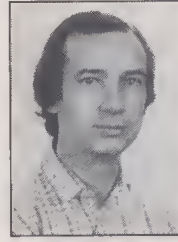
(4) Add new transactions. The backlog on some of our transactions is so long that it promotes the publication of archival materials in conference proceedings and commercial journals. We need to increase the number of transactions to meet our members' needs.

Thank you for reading my position statement.

Biography. Berra is currently an editor of *IEEE Transactions on Computers* and a lecturer on the Distinguished Visitor Program. In the past, he has served on the Board of Governors, as editor of *IEEE Transactions on Software Engineering*, as the editor-in-chief of the Computer Society Press, on the IEEE Press Editorial Board, as a member of the Distinguished Visitor Program, general chair and program chair of the International Conference on Data Engineering, and vice chair of the Publications Board.

He is currently a professor of electrical and computer engineering and a member of the faculty of Computer and Information Science at Syracuse University. He is also president of PBB Systems, an AI/database consulting firm. His industrial experience has included periods of service with Hughes, Bendix, and IBM. He previously held the position of professor and chairman of Industrial Engineering and Operations Research, also at Syracuse University. In addition to teaching at Syracuse, he has taught at the University of Michigan's Dearborn Campus, Boston University, and Purdue University.

Berra received the BS and MS degrees from the University of Michigan and the PhD from Purdue University. He is a fellow of the IEEE.



Paul Borrill

Position statement. The Computer Society goes from strength to strength. This, I believe, is due directly to the quality of service we provide to meet the

needs of our members in our publications, conferences, and tutorials and to the encouragement we provide for membership involvement in our chapter activities, technical committees, and standards activities. My interpretation of these needs is based simply on my own needs as a professional: to provide an environment in which to continue our education to nurture and develop our creativity, style, and professionalism in the most positive way possible.

This is precisely what my goals will be if elected as a member of the Board of Governors of the Computer Society.

Biography. First elected to the Board of Governors in 1984 as a petition candidate, Borrill has been active at various levels in the Computer Society: as the society's first ombudsman (a function he initiated from his concern over responsiveness to members), as secretary of the Board of Governors, as chair of the Audit Committee and the Standards Coordinating Committee, and as a member or chair of a wide range of Standards Working Groups, including the innovative IEEE P896.1 Futurebus approved as an IEEE standard in 1987.

Borrill is a staff scientist for National Semiconductor. Previously, he was head of R&D for Spectra-Tek. Earlier, as a research fellow at University College London, Borrill designed and programmed the flight control and ground-based data processing computer systems for the Coronal Helium Abundance Spacelab Experiment (CHASE), which flew in July 1985 on Spacelab 2.

Borrill received his BSc in physics (with honors) from the University of Manchester and his PhD in physics from the University of London. He is a member of IEEE and ACM and a chartered engineer with the IEE in the United Kingdom. Borrill has received NASA's Public Service Group Achievement Award and the Computer Society's Meritorious Service and Outstanding Contribution Awards.



Bill P. Buckles

Position statement.

The Computer Society is recovering from a financial crunch that crested two years ago. Its continued ability to deliver membership

services hinges on scrutinizing the ability of new activities to sustain themselves on a break-even basis. This applies primarily to conference and journal initiatives. The tutorial program—one of the more direct membership benefits—could be expanded within a cost-conscious framework. The long-term health of the society rests in recruiting and retaining new members. In this, a short-term investment/return analysis is not appropriate.

To help in recruiting and retaining new members, I support an expanded but cost-effective tutorial program and an expanded Distinguished Visitors Program, perhaps augmented by equivalent programs that are regionally administered. Within the context of this and other programs, I would seek ways to increase the advantages that regular chapters receive for assisting student chapters within their areas.

Biography. Buckles was newsletter editor for the Technical Committee on Distributed Processing (1979-84). He was an organizer and Steering Committee member for the First International Conference on Distributed Computing Systems and publicity chairperson for the third and fourth conferences. He was treasurer of the North American Fuzzy Information Processing Society (1985-87). Presently, he is a coeditor for a special issue of *Fuzzy Sets and Systems*.

Buckles is an associate professor of computer science at Tulane University. Previously, he was an assistant professor (later, associate professor) of computer science and engineering at the University of Texas at Arlington. For a number of years, he served in a variety of positions with Computer Science Corporation, Science Applications, and General Research Corporation providing technical and consulting support to NASA and the US Army Ballistic Missile Defense Command.

Buckles received a BS in mathematics from East Tennessee State University in 1967. He received an MS in operations research, an MS in computer science, and a PhD in operations research from the University of Alabama, Huntsville, in 1976, 1977, and 1981, respectively. He received the IEEE Outstanding Branch Counselor/Advisor Award in 1986. He is a Sigma Xi member and an IEEE senior member.



Jon T. Butler

Position statement.

The major challenge facing the Computer Society is delivery of critically needed services to a diverse community of professionals.

Traditionally, these services have included conferences and workshops for information dissemination among experts and tutorial sessions and texts for information dissemination from expert to layman.

I believe that the Computer Society's contribution to the computer community requires new initiatives. We must provide new services such as satellite conferences. And we must nurture emerging technologies with appropriate technical committees providing workshops, conferences, and newsletters. We must expand our tutorial series to include maturing fields.

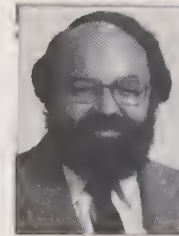
As a member of the Board of Governors, I will devote my energies to these goals, consistent with prudent use of our resources.

Biography. Butler has been active in the Computer Society since 1980. He has served as chair of the Multiple-Valued Logic Technical Committee, vice chair for hardware of the Technical Activities Board, member of the Distinguished Visitors Program, editor for *IEEE Transactions on Computers*, and editor for the Computer Society Press.

Since 1987, Butler has been a professor in the Department of Electrical and Computer Engineering at the Naval Postgraduate School in Monterey, California. He has served as the NAVELEX Chair Professor at the Naval Postgraduate School and on the faculty at Northwestern University. He has held summer positions in various companies, including Bell Telephone Laboratories in Naperville, Illinois.

He received the PhD degree in electrical engineering from Ohio State University in 1973 and BEE and M. Engr. degrees from Rensselaer Polytechnic Institute in 1966 and 1967, respectively.

Butler has been awarded the Meritorious Service Award and Certificate of Appreciation from the Computer Society. He has received the Outstanding Contributed Paper Award and the Award of Excellence of the Multiple-Valued Logic Technical Committee. He has been awarded two National Research Council Postdoctoral Associateships.



Michael Evangelist

Position statement.

Providing useful services must be the primary goal of the Computer Society. I will focus on two areas, publications and conferences:

(1) Magazines should balance the needs of experts with those of nonexpert professionals and students. We should ask each magazine to emulate *Computer* in publishing tutorials. Each should also analyze products and industry trends. For timely coverage of the most advanced technical subjects, we need at least two new transactions.

(2) Conferences should routinely sponsor new-product exhibits and tutorial sessions. Conferences should include sessions of brief tutorials, in addition to the current all-day tutorials. Interaction with colleagues is an important part of conference attendance. Too often, rigid schedules and formal meeting styles inhibit this interaction. Thus, at each conference, I want to see more small-scale roundtable discussions and better facilities for informal interaction.

These activities should devote significant resources to the professional needs of broad classes of membership.

Biography. Evangelist is an editor of *IEEE Transactions on Software Engineering* and a member of the Board of Governors, the Publications Board, and the Long-Range Planning Committee.

He previously chaired the Publications Planning Committee and the Magazine Advisory Committee. Evangelist has been a member of the *IEEE Software* Editorial Board and of several IEEE standards working groups. He has also served on the program committees of numerous conferences and has been a Computer Society distinguished visitor. In 1985, he helped launch *IEEE Expert*.

Evangelist leads the distributed computing project in the Software Technology Program at MCC. His group is developing a language, a methodology, and visual support tools for designing distributed and parallel systems. While at Bell Laboratories, he worked on a joint requirements-language project with the University of California, Berkeley, and did research in software metrics. He also spent four years as an assistant professor of computer science at Colgate University. He has published in distributed computing, software engineering, complexity theory, and mathematical logic.

Evangelist received his undergraduate degree from Beloit College and his MS and PhD in computer science from Northwestern University. He is a member of ACM and Phi Beta Kappa, in addition to the Computer Society.



Ronald G. Hoelzeman

Position statement. Technical excellence is the heart of the IEEE, and the Computer Society has been most aggressive in developing and providing new,

quality technical offerings for its members. As an active volunteer in the society, I have supported the continuing expansion of new services to meet the ever-broadening technical interests of the members. I have helped to launch four new magazines in response to this membership interest. In several other volunteer capacities, I have tried to keep expenses in line and to provide all of the society publications at the lowest member price. My goal as a board member will be to continue to promote new, quality technical member services at the lowest member cost. If elected, I will strive to be responsive to your needs and interests and will welcome your comments and ideas.

Biography. Hoelzeman has been involved in a variety of IEEE and Computer Society activities, primarily in the education and publications areas. He has served as the IEEE educational vice president and as a member of the institute's Board of Directors. For the Computer Society, he has served as a member of the Board of Governors and as treasurer. He has served on the society's Publications and Education Activities Boards, the Editorial Board of *Computer*, and was chairperson of the Magazine Advisory Committee. He is an ABET program evaluator for computer and electrical engineering programs, and is an alternate member of the ABET Board of Directors. He is one of the authors of the 1986 *Design Education in Computer Science and Engineering* and the Computer Society *Model Program in Computer Science and Engineering*.

Hoelzeman has been on the faculty of the Electrical Engineering Department of the University of Pittsburgh since 1970. Prior to that, he worked for the Westinghouse Electric Corporation.



Se June Hong

Position statement. I will focus positively on those areas where I feel strongly that our society needs improvement.

A technical society: The Computer Society is an international technical society, a professional organization. I believe all of its activities should focus on promoting its members' technical vitality and professional pride. This means responsible and high-quality professional output, vigorous technical exchanges, and emphasis on excellence in educational programs.

Quality: As a member of the Board of Governors, my first concern will be to see that the society adheres to a high standard of quality that is reflected in our transactions, journals, magazines, and conferences and in the leadership positions we take.

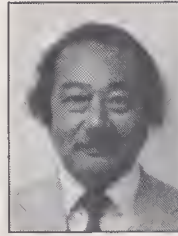
Technical committees: I believe our various technical committees are the basic building blocks of our society. I would encourage every society member to participate in specialty-area technical committees and to use that forum for timely and active technical exchanges.

Biography. An IEEE fellow, Hong has been a Computer Society distinguished visitor, treasurer of a Fault-Tolerant Computing Symposium, and a member of the IEEE/EAB Ad-Hoc Visiting Committee on Accreditation and the IEEE Edison Medal Committee. He has also served on the program committees of several artificial intelligence conferences and workshops.

He has guest edited two special issues—the July 1982 *IEEE Transaction on Computers* on fault-tolerant computing and the July 1986 *Computer* on expert systems for engineering applications—and has served on the *Journal of Automated Reasoning* and *New Generation Computing* Editorial Board since 1984.

Hong has been a senior manager responsible for the knowledge systems area at IBM's T.J. Watson Research Center since 1983. He has been with IBM since 1969 and has worked on a parallel machine for chip wiring, error-correcting codes, test algorithms, and PLA design algorithms, including MINI for minimizing large switching functions.

Hong received his BSEE from Seoul National University in 1965 and his MS and PhD in electrical engineering from the University of Illinois in 1967 and 1969. He has received five IBM Invention Achievement Awards for 16 US patents, four IBM Outstanding Innovation Awards, an Eta Kappa Nu honorable mention (1975 Outstanding Young Electrical Engineer), and a New York State Jaycees Distinguished Service Award.



Tadao Ichikawa

Position statement. The Computer Society has made remarkable progress in recent years. There are more members, more publications, and a wide range of professional activities. That's very encouraging, but we'll have to ensure that it doesn't unintentionally result in a more hierarchical, less accessible structure.

The topics we focus on should not be chosen solely because they reflect majority interests. Individual members should always be encouraged to initiate new activities. Cooperation and interaction on these new areas of research should not be limited to any particular section of the society. We need, therefore, a framework that will encourage and improve interaction and communication between members and governors in every branch of the society, regardless of position or geographical location.

If elected, I will work toward establishing simpler, more efficient means of communication designed to promote innovative contributions from individual members. This will make the society more beneficial to everyone.

Biography. Ichikawa has been active in program organization as a Program Committee member for the IEEE International Conference on Data Engineering and many other IEEE conferences, symposia, and workshops. In 1984, he organized the new IEEE Visual Languages Workshop. He was general chairperson of the second workshop in 1986 and is currently serving as a member of the workshop Steering Committee. He also serves as a member of the Asian Activities Committee and is on the Editorial Board of *IEEE Transactions on Software Engineering*.

Ichikawa is a professor of computer engineering at Hiroshima University in Japan. His research interests include database systems, visual programming, and software development environments. Prior to his professorship at the university, he worked in the private sector where he made major contributions to the development of computerized satellite communication systems in the early stages of computer utilization in Japan.

Ichikawa received his BE degree in 1962 and doctor of engineering degree in 1971, both from Waseda University, Tokyo. He received the Yonezawa Memorial Science Promotion Award and the Best Paper Award from the Institute of Electronics, Information, and Communication Engineers of Japan in 1967 and 1980, respectively. He is a fellow of the IEEE.



Sushil Jajodia

Position statement.

"The greatest thing in the world is not so much where we stand as in what direction we are moving."

—Oliver Wendell Holmes

The Computer Society must be a force for the well-being of our profession. Since people are the source of new knowledge, we must develop human resources to meet the demands of the future. We must provide better and more services at every level: publications reflecting the evolving nature of computer technology, conferences for exchanging ideas on current research, and tutorials on theoretical and practical topics. Also, the Technical Activities Board must be vitalized so that its technical committees are more visible to members.

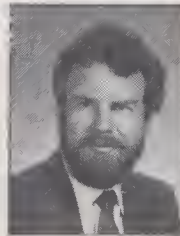
I began my volunteer career chairing the Technical Committee on Data Engineering two years ago. These were years of growth, both personal and professional. The TC benefited also from my active, innovative, and effective leadership. With your support, I will put these skills to work as a member of the Board of Governors.

Biography. Jajodia is a member of the Publication Planning Committee and the TAB Operations Committee. As chair of the Technical Committee on Data Engineering, his entrepreneurial efforts were successful in obtaining the financial support to meet the publication cost of the *Bulletin on Data Engineering*. He also played a vital role in developing the proposal for the new *IEEE Transactions on Knowledge and Data Engineering* and securing its approval by the Computer Society.

Jajodia has served in different capacities for various journals and conferences. He is currently serving as program cochairperson of the Fifth Data Engineering Conference and the ACM/IEEE Databases in Parallel and Distributed Systems Symposium.

Jajodia is an associate professor in Information Systems and Systems Engineering, George Mason University, Fairfax, Virginia. Before that he was program director at the National Science Foundation and headed the Database and Distributed Systems Section at the Naval Research Laboratory, Washington, DC. He has published over 40 research articles and coedited a book.

He received his PhD in mathematics from the University of Oregon. He is a senior member of IEEE and a member of ACM. He received a Computer Society Certificate of Appreciation for his contributions to the Fourth Data Engineering Conference.



Ted Lewis

Position statement.

The Computer Society has been very successful in the past because of its responsiveness to the needs and desires of its members. We must

continue to provide timely, accurate, and useful information to our members through support of professional activities, publications, and tutorials.

I am especially interested in continued improvement in our publications (for example, *Computer*, *IEEE Software*, *IEEE Expert*, and the various tutorial books). As a past coeditor of *Computer* and the current editor-in-chief of *IEEE Software*, I have participated in the transition of these publications from limited to widespread readership.

Our publications have become widely referenced in other journals. We now need to find ways to make them available to a broader group of practitioners. We must enable accessing by students, engineers in other disciplines, scientists, and educators to further the use of computing in their respective occupations.

Biography. Ted Lewis is the editor-in-chief of *IEEE Software*. He was previously a co-editor of *Computer*, for which service he received the Computer Society Honor Roll Award in 1982.

Currently a professor of computer science at Oregon State University, Lewis has been on the computer science faculties at the University of Southwestern Louisiana (1973-76) and the University of Missouri-Rolla (1971-73). He is the coinventor of the Tausworthe-Lewis-Payne algorithm (used extensively in cryptography) and is the developer of the first retargetable microprogramming language system for implementing efficient microprograms.

Lewis has published extensively in the areas of software engineering, distributed operating systems, and computer organization and architecture; he has authored a number of books on personal computers, software engineering, and Pascal.

Lewis received the BS degree in mathematics from Oregon State University in 1966 and MS and PhD degrees from Washington State University in 1970 and 1971, respectively.



Ming T. (Mike) Liu

Position statement.

A constant challenge facing the computer professional is how to keep pace with rapidly advancing technologies such as work stations,

distributed computing, supercomputing, and neural networks. The main purpose of the Computer Society is to provide technical programs and professional services to enable its members to keep abreast of technological changes. The society serves its members through publications, conferences and tutorials, chapter and educational activities, and technical and standards activities. All of these functions must remain active, strong, flexible, and innovative so as to maintain the technical vitality and leadership of its members.

In the past, I have strongly supported and actively participated in these activities. If elected, I will continue to support these activities and to seek innovative and effective ways to help our members successfully meet their professional challenges.

Biography. Liu currently serves as editor-in-chief of *IEEE Transactions on Computers* and as a member of the Board of Governors, the IEEE Fellow Committee, and the Constitution and Bylaws Committee. He also serves as faculty advisor to the Ohio State University student chapter.

In the past, Liu served as vice president for membership and information and as chairperson of the Tutorials Committee, Elections Committee, Technical Committee on Distributed Processing, and the ACM/IEEE Eckert-Manchly Award Committee. He also served as program chair and general chair of the International Conference on Distributed Computing Systems in 1985 and 1986, respectively. He received Meritorious Service Awards in 1985 and 1987 for his contributions to membership, tutorials, conferences, and publications.

A professor of computer and information science at Ohio State University, Liu has published over 100 technical papers and supervised over 30 doctoral dissertations. In 1983-84, he spent his sabbatical leave at AT&T Bell Laboratories. In 1983, he was elected an IEEE fellow. In 1987, he received the Distinguished Achievement Award from his alma mater and the Outstanding Member Award from the IEEE Columbus Section.

Liu holds MSEE and PhD degrees from the University of Pennsylvania. He is also a member of ACM and Sigma Xi.



Raymond E. Miller

Position statement. The Computer Society needs to strengthen its public image as an authoritative body speaking for computer science and engineering.

Computer professionals should have a major, unified impact on how society views and uses computing technology, and the Computer Society should play a dominant role in improving society's view.

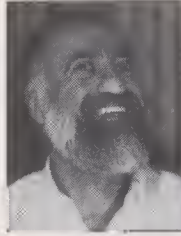
The basis for this role has already been established through the Computer Society's membership strength and quality programs in publications, conferences, and education. These programs must remain strong, while still being open to growth and change as our dynamic field evolves.

I have been an IEEE and Computer Society member since the 1950s and an IEEE fellow since 1970. I have served the Computer Society, the ACM, the Computing Sciences Accreditation Board, the Computing Research Board, and AFIPS in various roles. Thus, I am well qualified and look forward to board service aimed at improving the Computer Society's activities and role in society.

Biography. Miller chaired the Technical Committee on Switching and Automata Theory from 1969 to 1972 and has been a member of the Fellows Committee since 1984, serving as chair in 1986 and 1987. In ACM, he was editor-in-chief of the *Journal of the ACM* (1972-76), a member of the ACM Publications Board (1973-87), and a member-at-large of the ACM Council (1976-82). He served on the AFIPS Board of Directors and was an initial member of the CSNet Executive Committee. He served as vice president and president and is currently past-president of the Computing Sciences Accreditation Board.

In July 1988, Miller became director of the NASA Center of Excellence in Space Data and Information Sciences. With this position, he will relocate in the spring to the University of Maryland, College Park, where he will be a professor in the Computer Science Department. Currently, Miller is a professor in the School of Information and Computer Science at Georgia Institute of Technology; he was the school's director from 1980 to 1987. From 1957 to 1980, he was a research staff member at IBM's T.J. Watson Research Center.

Miller earned bachelor's degrees in mechanical and electrical engineering, a master's degree in mathematics, and a PhD in electrical engineering from the University of Illinois. He has received several IBM invention awards and a Computer Society Outstanding Contribution Award.



Yale N. Patt

Position statement. I am a candidate for the Board of Governors because I believe a strong professional society—governed by its members, committed

to quality, and uncompromising with respect to fads, hype, or quackery—is a good thing. That means that, if elected, I will agree to serve and that I will focus my energies in two areas: the technical quality of our publications and the legitimacy of our accreditation processes. I believe that the Computer Society has a special mission with respect to accreditation: responsibility for participating in the validation of the academic programs that turn out our graduates. In my view, that responsibility is primarily to the profession and the society that use these graduates, and very much less so to the universities that produce them. I will work toward meeting that responsibility.

Biography. A member of the *Computer Editorial Board*, Patt is guest editor of a special issue planned for January 1989 and titled, "Real Machines: Design Choices/Engineering Trade-offs." He is chairing Micro 21, the Workshop on Micro-programming and Microarchitecture to be held in late November, and served on the Program Committee of the four preceding Micro workshops. He was program cochair of the 1988 International Computer Architecture Conference and has served on the Program Committee of every Compcon Spring since 1984.

For the past 12 years, Patt has been involved in computer architecture research and teaching at the University of California, Berkeley, and at San Francisco State. He is also a consultant to DEC, NCR, and Nexgen on problems relating to high-performance computer implementation.

Patt received his BS degree from Northwestern University and his MS and PhD degrees from Stanford University, all in electrical engineering.



Earl E. Swartzlander, Jr.

Position statement. For two decades, I have served the Computer Society in a variety of roles. Especially in the last

two years, as a member of the Board of Governors, I have learned much about its inner workings. I hope to be reelected so that I may continue to serve you.

The society is now on a firm financial foundation, with strong journals, magazines, books, and conferences, but we can improve the coupling between technical activities, especially between the Technical Activities Board and our conferences.

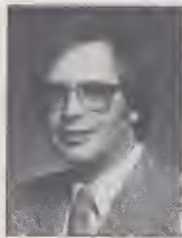
Although strong, our publications program can be improved. We need additional transactions to relieve the pressure on existing transactions, and we can harness the desktop-publishing revolution to reduce magazine production costs while improving technical vitality.

In working with the Computer Society Press, I helped initiate hardbound conference proceedings to give authors and attendees a tangible benefit. We can expand this to more conferences, symposia, and workshops.

Biography. Swartzlander is an associate editor of the *IEEE Journal of Solid-State Circuits*, general chair of the International Conference on Wafer-Scale Integration, and a member of the Computer Society's West Coast Operations Committee, Constitution and Bylaws Committee, and Technical Activities Board. He is a CS representative to the IEEE Solid-State Circuits Council. He is also the hardware area editor of *ACM Computing Reviews* and editor-in-chief of the *Journal of VLSI Signal Processing*, which will begin publication in 1989.

As director of independent research and development for TRW's Defense Systems Group, Swartzlander is responsible for defining, implementing, and directing an integrated program for hardware, software, and systems research and development. Previously, he held a variety of technical, line, and staff positions within TRW, Hughes Aircraft, and Ball Aerospace. He authored the book *VLSI Signal Processing Systems*, edited *Computer Arithmetic* and *Systolic Signal Processing Systems*, and has written 100 papers on computer arithmetic and VLSI architecture.

Swartzlander earned his BSEE at Purdue University, his MSEE at the University of Colorado, and his PhD at the University of Southern California. He is an IEEE fellow, a member of Eta Kappa Nu, Sigma Tau, and Omicron Delta Kappa, and a registered professional engineer in Alabama, California, and Colorado.



Donald E. Thomas

Position statement.

One of the roles of the Computer Society is to promote worldwide technical interchange. The CS fulfills this role through its extensive publishing, conference, and standards activities. We must continue to upgrade these member services while keeping them economical.

Conferences and publications are a major means of transferring technical information. I would promote activities supporting topical conferences, and promote journal and magazine publications as reader interest dictates.

The Compmail facility has become a great asset in helping our society communicate. We must continue to increase its capabilities, tying it into other networks. This will more closely integrate those members who already have established network connections.

The future direction of the CS is driven by the volunteers, people who, with the support of their employers, provide dynamic leadership for the society's activities. We must continue to develop a staff that will be supportive of this leadership.

Biography. Thomas is a senior member of IEEE, a member of ACM and Sigma Xi, and chair of the 26th Design Automation Conference. He served as program chair of the 23rd and 24th Design Automation Conferences. He was a founding editor of *IEEE Design & Test* and served as guest editor of the August 1984, August 1985, and February 1987 issues of *D&T*. He is a member of the society's Technical Committee on Design Automation and of ACM's SIGDA.

Thomas is currently a professor of electrical and computer engineering at Carnegie Mellon University and associate director of the SRC-CMU Research Center for Computer-Aided Design. While on sabbatical leave in 1985-86, he was a visiting scientist at the IBM T.J. Watson Research Center. His research interests include the automatic design of integrated circuits and systems, where he has developed and demonstrated techniques for synthesizing register-transfer-level designs from program-like behavioral descriptions. He has published in numerous journals and conference proceedings.

He received his PhD degree in electrical engineering in 1977 from Carnegie Mellon University. He received the society's Meritorious Service Award for contributions as editor of *D&T* and the *IEEE Transactions on Education* Outstanding Paper Award for 1978-1979.

He received his PhD degree in electrical engineering in 1977 from Carnegie Mellon University. He received the society's Meritorious Service Award for contributions as editor of *D&T* and the *IEEE Transactions on Education* Outstanding Paper Award for 1978-1979.



Jacques Tiberghien

Position statement.

The Computer Society has the best potential for growth in Regions 8 and 10. These are technologically advanced regions where only a

small percentage of computer professionals are Computer Society members. The Computer Society's offices in Brussels and Tokyo increase society visibility, but a real membership breakthrough requires more than a successful commercial operation. The Computer Society's transnational character has to be emphasized through good contacts with regional societies and through "regional" directors on the Board of Governors.

If elected, I will have sufficient credibility to further develop friendly relations with various European societies. I will invite those of their influential volunteers who are also Computer Society members to become actively involved in CS activities.

To develop—in cooperation with local societies—CS technical activities in Europe, I will try to enable and motivate the Brussels office to effectively support such activities.

Biography. Tiberghien has been active in the Computer Society, first as a member and now as the chair of the European Activities Committee. As such, he was instrumental in the growth of the European Office in Brussels. He was also the general chair of CompEuro 88, the annual conference jointly organized by the Computer Society and IEEE Region 8. He was awarded the Computer Society's Certificate of Appreciation for the role he played in the development of its European Office.

Since 1977, in addition to his Computer Society activities, he has been a director of EuroMicro and chair of the Committee on Informatics of the KVIV, the largest Belgian engineers society.

A full professor at the Engineering School of the Vrije Universiteit Brussel (Free University of Brussels), Tiberghien teaches introductory courses in computer sciences and more specialized courses on computer architecture and computer networks. He is vice chair of the Board of Directors of the computer center common to the two Free Universities in Brussels. He has consulted for several large organizations and, during sabbaticals, has visited at Dalhousie University in Nova Scotia and the University of California at Berkeley. He has authored one textbook and over 40 scientific papers and has edited several conference and professorship proceedings.

Tiberghien received his electrical engineering degree from the Universit Libre de Bruxelles in 1969 and his PhD from the Vrije Universiteit Brussel in 1976.



Mario Tokoro

Position statement.

Expansion in quantity and improvement in quality of a scientific society tend to occur alternately. The Computer Society has

successfully attracted a wide range of people in computer science, engineering, manufacturing, and education. Now is the time to promote the society's scientific and technological activities. I intend to devote myself to broadening the society's perspectives and, at the same time, deepening its scientific and technological strength in each spectrum.

Biography. Tokoro served as vice program chair of the 10th International Symposium on Computer Architecture in 1983 and as program chair of the 13th International Symposium on Computer Architecture in 1986. Currently, he is serving as a referee for *Computer*, *IEEE Software*, and two of the Computer Society's transactions, as well as serving on the Editorial Board of the Japan Society of Software Science and Technology and on IFIP Working Group 10.1.

He is an associate professor in the Department of Electrical Engineering, Keio University, Yokohama, Japan, and a director of Sony Computer Science Laboratory. He was a visiting assistant professor at the University of Waterloo from January 1979 to June 1980 and at Carnegie Mellon University from July to December 1980.

His recent research accomplishments include development of a microprogram optimization technique, a multiprocessor system with local and shared memory, the object-oriented architecture Zoom, and the object-oriented concurrent programming languages Orient 84/K and Concurrent Smalltalk. His current interests include object oriented programming languages/systems and distributed operating systems.

Tokoro received a BS in electrical engineering, an MS in administration engineering, and a PhD in electrical engineering from Keio University in 1970, 1972, and 1975, respectively.



Benjamin W. Wah

Position statement. Today's application-oriented computer science and engineering fields present two urgent challenges to the Computer Society.

One is the cross-fertilization of ideas with other professional societies that address research problems of interest to our members. A mutual exchange could benefit all parties but would require closer cooperation in sponsoring conferences and other technical activities. If organized effectively, this cooperation could benefit the Computer Society both technically and financially.

The second challenge concerns the magazines, journals, and conferences already sponsored by the Computer Society. Review procedure, turnaround time, and quality differ for each publication. Authors, editors, guest editors, reviewers, and readers would benefit if procedures for handling papers were more uniform and readily available to members and other interested parties.

I have ideas and experience to contribute in both of these areas, and would be honored to work with other board members to strengthen and expand current policies and programs.

Biography. Wah is the associate editor-in-chief of the forthcoming *IEEE Transactions on Knowledge and Data Engineering* and an editor of *IEEE Transactions on Software Engineering*. He also serves on the boards of three other publications and has served as a guest editor for *Computer* and for *IEEE Transactions on Software Engineering*.

Wah was the general chair and program chair of the IEEE International Conference on Data Engineering, and a program vice chair for two IEEE International Conferences on Distributed Computing Systems. He has also served on the program committees of 14 other conferences.

Wah is an associate professor in the Department of Electrical and Computer Engineering and a research associate professor in the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. He is currently on leave at the National Science Foundation as a program director in the Microelectronic Information Systems Division. His areas of interest include computer architecture, parallel processing, artificial intelligence, distributed databases, and operating systems. He has authored over 100 articles in those areas.

An IEEE senior member and a Computer Society distinguished visitor, Wah received his PhD in computer science from the University of California at Berkeley.



Ronald Waxman

Position statement. An interesting aspect of being part of the Board of Governors is the chance to affect the way we go about increasing technical

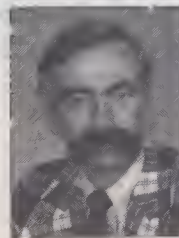
vitality. I will work to find new and innovative ways to foster the interest of computer professionals in joining the Computer Society. Those ways must strike a responsive chord in prospective members. In my experience with Computer Society standards activities, I found that direct involvement in making things happen drew me into the fold. Leadership possibilities abound within the Computer Society, and more leaders bring more activities, which lead, in turn, to more involved members.

Leadership opportunities include initiating international technical exchange, standards-making activities, technical workshops into areas not yet explored, more special issues of the various Computer Society magazines, and an open forum for ideas from members and prospective members. My efforts will be directed to making those opportunities visible and encouraging active participation.

Biography. Waxman is chair of the Technical Committee on Design Automation, editor of "D&T News" for *IEEE Design & Test*, and an active member of various Computer Society boards. He founded the Design Automation Standards Subcommittee (DASS) and represented the Computer Society on the IEEE NESCOM, NOSCOM, and ATLAS Committees. He has organized and chaired technical functions for the Computer Society and other professional organizations and guest-edited a special issue of *D&T*.

A principal scientist at the University of Virginia's Center for Semicustom Integrated Systems since 1987, Waxman conducts research in the design of digital systems. He was employed by IBM for 32 years in technical and management capacities, attaining the position of senior engineer. He currently consults for IBM and the government. At IBM, he guided the development of VHDL which, under his DASS leadership, became an IEEE Standard in 1987. He has numerous patents and publications.

Waxman received a BSEE from the New Jersey Institute of Technology and an MEE from Syracuse University. An IEEE senior member and Computer Society distinguished visitor, his honors include a Computer Society Meritorious Service Award, IBM awards for technical achievement, and membership in Sigma Xi, Eta Kappa Nu, and Phi Eta Sigma.



Thomas W. Williams

Position statement. If I am elected to the Board of Governors, I will strive to bring a balance between the objectives of the Computer Society's

industrial and academic segments. I will also work to ensure very high quality in the technical programs I am involved in and any others that come before the board for discussion and/or action. My final objective is to utilize new technologies—for example, satellite TV—to disseminate information to our 90,000 members. However, I do not believe in using technology for technology's sake and will work to maintain the areas of the society that are operating in good order.

In summary, my objectives are to

- balance the objectives of industry and academia;
- ensure very high quality in our conferences, workshops, and meetings;
- utilize new technologies in areas that need improvement; and
- maintain areas that are in good operating order.

Thank you for your consideration.

Biography. Williams is the founder and chair of the Computer Society's Workshop on Design for Testability, which recently had its 11th annual meeting in Vail, Colorado. He also chairs the Technical Subcommittee on Design for Testability and has represented the society on the CompEuro Steering Committee. He has been on the Program Committee of many testing conferences, as well as being the keynote and invited speaker at a number of conferences both in the US and abroad. He was a Computer Society distinguished visitor from 1982 to 1985.

Williams is a senior technical staff member at the IBM Systems Technology Division in Boulder, Colorado. He manages the VLSI Design Rules Group, which deals with design for testability of IBM products. He is also an adjunct professor at the University of Colorado, Boulder, and has been a guest professor at Universitaet Hannover.

He received a BSEE from Clarkson College of Technology, an MA in pure mathematics from the State University of New York at Binghamton, and a PhD in electrical engineering from Colorado State University.

Williams has received three patent awards from IBM and a number of best paper awards, the most recent from the International Test Conference for his work in VLSI self-testing.

UPDATE

Contributions to Update are welcome. Send news of industrial or university research and of public policy or professional issues to Update Editor, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720, or to Bruce Shriver, Editor-in-Chief, Dept. of Decision Sciences, University of Hawaii, 2404 Maile Way, Honolulu, HI 96822.

Computer industry experts put minds on the line

Who sold Katherine Hepburn a computer in the movie "Desk Set"? Which computer entrepreneur got his first venture capital as the result of a dog bite?

Such technotrivia will be the order of the day as industry experts test their knowledge of computer history, technology, business, and folklore at the world's first Computer Bowl. The contest, sponsored by the Computer Museum, will be held October 7 at Boston's World Trade Center.

Among those pitting their wits against one another are David Bunnell, chairman and CEO of PCW Communications; Esther Dyson, editor-publisher of the newsletter *Release 1.0*; David Hathaway, partner in Venrock Associates; William "Bill" Joy, cofounder of Sun Microsystems; Mitchell Kapor, chairman of On Technology and founder of Lotus Development; Allen Michels, president of Ardent Computers; John William Poduska, chairman and CEO of Stellar Computer and founder of Prime and Apollo Computer; Casey Powell, president and CEO of Sequent Computers; and Richard Shaffer, editor-publisher of the *Technologic Computer Letter*.

William Randolph Hearst III, editor-publisher of the *San Francisco Examiner*, will be among those posing the questions, and Mike Perkowski, former "Back Page" writer for *Computer Systems News*, will be one of three judges. (As of press time, the identity of the celebrity Bowl host remained a mystery.)

The location of the next Bowl (to be held in 1990) will be determined by this year's winners, who will co-sponsor the event with the Computer Museum.

For more information about the Computer Bowl, call the Computer Museum at (617) 426-2800.

IEEE urges commitment to space program

The IEEE has joined the American Astronomical Society and the American Institute of Aeronautics and Astronautics in a public statement calling for "the leadership of the United States to reverse this nation's retreat from exploration and development in space by providing adequate resources to NASA."

"The United States civilian space program is at a crossroads," the societies said in the July 22 statement. "Its future is being debated in Congress, at NASA, and among the men and women preparing the transition to a new Administration."

"We believe NASA's program is neither sufficiently understood nor adequately supported in this critical period," the statement continued. "It is our responsibility to call to the nation's attention the great value of NASA's historical contributions to education, science, exploration, and leadership. We urgently recommend restoring national commitment to purposeful

civilian space programs and policies."

"The problem is that budgetary growth is required through the remainder of the century to exploit the technology and infrastructure created by NASA. That is the conclusion of the May 1988 report by the Congressional Budget Office entitled *The NASA Program in the 1990s and Beyond*.

"Our nation's most conspicuous endeavors must be successful, as we are a prideful people. Inadequate support tempts failure. We have depleted and we must now restore the scientific and technical capital gathered in NASA's early years. New generations of young people must be attracted to educations in engineering, science, and mathematics. Innovations in commercial products and processes must be found to maintain our competitiveness in the world marketplace."

The statement was signed by IEEE President Russell C. Drew, AAS President Donald E. Osterbrock, and AIAA President William F. Ballhaus, Jr.

NSF announces women's professorship program

The National Science Foundation has announced a November 15 deadline for applications to its program of visiting professorships for women.

The program is designed to aid career advancement, offer greater visibility for women scientists and engineers, and encourage other women to pursue careers in science and engineering.

Awardees are responsible for selecting and making all arrangements with the host institution. Proposals will be judged on the scientific merit of the proposed research and the quality of plans for lecturing and other interactive activities.

Eligible candidates must hold a doctorate in a research field supported by NSF (or have equivalent experience);

have independent research experience in the academic, industrial, or public sectors; be currently or recently affiliated with a US institution; and not have a salaried position or the promise of one with the host institution at the time of application.

Awards may range from one academic semester to 24 months, full- or part-time. Award announcements will be made June 15, 1989, with the earliest starting date being July 1, 1989 and the latest starting date being July 1 1990.

Single copies of the program announcement are available from the Forms and Publications Unit, Room 232, National Science Foundation, Washington, DC 20550.

ER APPROACH

November 16-18, 1988

7th International Conference on Entity-Relationship Approach

Rome, Italy

Hotel Ambasciatori

Presented by: In Cooperation with:

THE
ER
Institute



ACM
SIGBDP



IEEE
Computer
Society



Data Administration
Management
Association

afcet
AICA

IASI-CNR
GI

Sponsored by:

With the patronage of:

ITALSIEL - TECSIEL (FINSIEL-IRI group)

MINISTERO DELLA RICERCA SCIENTIFICA - ITALY

ISTITUTO CENTRALE DI STATISTICA - ITALY

PROVVEDITORATO GENERALE DELLO STATO - ITALY

Coordinated in Italy by:

GESI Srl

The Entity-Relationship Approach is the basis for many Database Design and System Development Methodologies. ER Conferences bring together practitioners and researchers to share new developments and issues related to the use of the ER Approach.

3-DAY CONFERENCE FEATURES:

- * Internationally Known Experts in Database and Information Management from both Academic and Practitioner Communities
- * Tutorials, Technical Presentations, Panel Discussions, Vendor Demonstrations * Conference Proceedings and Tutorial Materials
- * Traditional Italian Banquet * Lunches and Continuous Coffee Service Provided Each Day of the Conference

PROGRAM HIGHLIGHTS:

Papers Presented on Major Themes:

Database Development and Management

- * database design
- * database management systems
- * languages for data description and manipulation

Application Systems

- * CAD/CAM and engineering databases
- * knowledge-based systems
- * object-oriented systems

Managing Organizational Information Resources

- * data planning
- * information architectures
- * translating data plans into systems

- * data models and modeling
- * database dynamics
- * database constraints

- * multi-media databases
- * user interfaces
- * business systems

- * data dictionaries
- * information centers
- * end user computing

Tutorials Include:

- * Klaus Dittrich
Universitat Karlsruhe (West Germany)
Object Oriented Databases
- * Stefano Ceri
Politecnico di Milano (Italy)
Distributed Database
- * Gianfranco Potevolpe
Italsiel (Italy)
Software Maintenance
- * Sham Navathe
University of Florida (USA)
Conceptual and Logical Database Design

Conference Location:

The Conference will be held at the Hotel Ambasciatori, a First Class Hotel located at Via Veneto 70, Rome, Italy 00817 (396-47493). Via Veneto is among the most touristic streets in Rome. More moderately priced hotels, convenient to the conference include: Hotel Savoy, Via Ludovisi 15 (396-4744141), Sicilia Hotel, Via Sicilia 24 (396-493841). Less expensive hotels include: Hotel Veneto, Via Piemonte 63 (396-789251), Hotel Alexandra, Via Veneto 18 (396-481943).

CONFERENCE FEES:

	Prior to Sept. 25	After Sept. 25
___ Regular	\$375.00	\$425.00
___ Discount	\$325.00	\$375.00
___ Student	\$100.00	\$150.00

- * Discount applies to Cooperating and Affiliated Organizations
- * Student Fee Excludes Luncheons and Banquet

U.S. REGISTRATION:

Professor Salvatore T. March
Department of Management Sciences
University of Minnesota
271 19th Avenue South
Minneapolis, MN 55455 USA
(612) 624-2017; march@umnacvx.bitnet

NAME _____ Discount Organization _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____ PHONE _____

Make Checks Payable to: Seventh International ER Conference (US Dollars Only)

NEW PRODUCT REVIEWS

Editor: Richard Eckhouse, MOCO, Inc., PO Box A, 91 Surfside Rd., Scituate, MA 02055; Compmail+, r.eckhouse

Ada on the IBM PC—it's for real!

T. L. (Frank) Pappas, Intermetrics

Ada compilers are now available for 80x86 systems running either PC-DOS or MS-DOS. This makes Ada available to a much larger group of users. But if you're like most people who make the move to Ada, you really don't have any idea how to select an Ada compiler. Sure, you may have selected the C, Fortran, or Pascal compiler you're using, but the issues in picking an Ada compiler are somewhat different. If you just plunge in and pick a compiler, you stand a good chance of picking the wrong one. Then there's the question, how do you or your development team learn to use Ada? What about tools that can make your move to Ada a little bit easier?

In this review, we'll look at the three Ada compilers available under DOS and at a few of the issues you need to keep in mind when picking an Ada compiler. We'll also look at two tools that may help you. One is an on-line Ada Reference Manual (RM). The other is a Pascal-to-Ada translator. In a future issue, we'll look at some computer-based Ada training products.

Why move to Ada at all?

As most of you probably know, Ada's development was sponsored by the United States Department of Defense (DoD). The idea was to have a single high-level language that supported modern software engineering principles and could be used to develop complex military systems.

One of the curious aspects of Ada's development was that early Ada compilers weren't adequate for such DoD systems and couldn't be used as the DoD expected. On the other hand, the commercial (non-real-time) market had

good Ada compilers, but viewed Ada as a DoD aberration and stayed away from it. Today the situation is changing. Quality Ada compilers for complex military and commercial systems are now available. And they're being used successfully.

Commercial use will grow more rapidly now that Ada is so easily available on the PC. Many small companies are already turning to Ada as a way of reducing their development costs. For example, two companies that I know of have rewritten their commercially available database management systems in Ada.

Moreover, you can expect vendors of the three compilers reviewed here to offer full 80386 implementations in the very near future. These vendors will even offer Ada under Unix 386-based systems, some by the end of the year. I hope to let you know about these compilers in a follow-up review.

Why would anyone want to use Ada in the first place? For this, we have to look at why the DoD sponsored Ada. Typical of the systems for which the DoD wanted Ada are sophisticated embedded weapons systems. With such systems, a malfunction could easily result in loss of life. This meant an emphasis on reliability as one of Ada's main design goals. Since the DoD probably has the world's largest maintenance headache, readability, adaptability, portability, and reusability were also emphasized.

But the very features that help in writing these military systems also make Ada a superior language for commercial software domains. Reliability is just as important when performing financial transactions, monitoring patients in an intensive care unit, or monitoring aircraft in an air traffic control system. Economic factors also make the other

aspects of Ada, such as adaptability and reusability, important in these domains.

As a side benefit of the DoD sponsoring and standardizing Ada, there's a strong economic incentive for software vendors to develop Ada products. These vendors know that a large marketplace exists for Ada products. Commercial (non-DoD) users will benefit from the availability of these tools as well.

And then, consider all that free Ada software. That's right, free Ada software! To promote the use of Ada, the Ada Joint Program Office (AJPO) set up an Ada software repository (see the sidebar "Ada Information Clearinghouse"). You can dial up the repository and get some of the software through file transfers, but you're probably better off paying the cost of a tape because of the quantity. Not all of it is great software, and much of it seems to have been written for DEC's Ada compiler, but it shouldn't be too difficult to modify.

Finally, the DoD has given out R&D contracts for reusable Ada software under its STARS (Software Technology for Adaptable, Reliable Systems) program. This software, from various application domains, will also be available free.

So there seems to be a lot of strong economic incentives for moving to Ada.

First a word of caution. Whenever you read or hear something about Ada, always consider the source. Too much misinformation about Ada already exists, both pro and con.

For example, a well-known defense industry magazine contained an article on Ada and C³I. The author was listed as his company's "Ada expert." This "expert" stated that Ada runtime checking can be quite inefficient. To illustrate his point, he used an integer

type named Prime as if it were predefined in Ada. He then went on to explain that Ada runtime checking had to ensure that only prime numbers could be assigned to Prime variables. I wonder if the company this "expert" worked for is still in business?

Anyway, at the other extreme, some advocates of Ada seem to have had an almost religious experience with it. To them, the language has no flaws and all of the compilers are great.

You often hear people argue about the merits of Ada versus C or C++ . If you listen carefully to these arguments, you'll find a strong parallel to those used years ago for Fortran and assembly language. Assuming reasonable compilers for the same target computer, there's really no technical reason why C could be used on one project while Ada couldn't, and vice versa. The software engineering aspects make Ada more attractive.

Based on what I just wrote, it's only fair that you should know my perspective on Ada. For more than five years, I have looked at Ada from several different points of view. I was involved in the development of the Army's Ada Training Curriculum and have taught Ada programming courses many times. I have done studies on Ada portability and reusability, and am currently writing a book on Ada reusability. I have done a fair amount of Ada programming to boot. I think I have a pretty balanced view of the language.

I don't think Ada is the perfect programming language. But overall, I think it's a very good software engineering language, and it's the best that will be available to us on a large scale for quite some time.

Now let's take a look at Ada on the PC.

Ada compilers

When choosing an Ada compiler, you should stick to one that's validated, unless you know what you're getting into. If you deliver an Ada program to a DoD organization, the program must compile under a validated compiler and execute under the compiler's runtime system.

A validated compiler means that the AJPO certifies that the compiler has successfully passed the tests in the Ada Compiler Validation Capability (ACVC) testbed. The ACVC contains hundreds of tests to check that the compiler conforms to the RM.

The AJPO validates conformance based on these tests. This doesn't mean

Ada Information Clearinghouse

You can get information about validated Ada compilers, the Ada software repository, and other Ada-related information by writing to the

Ada Information Clearinghouse
3D139 (1211 S. FERN, C-107)
The Pentagon
Washington, DC 20301-3081

or calling them at either

(703) 685-1477
or
(301) 731-8894

that a particular compiler conforms in some area not tested by the ACVC. But new tests are added each year, based on interpretations from the Ada Board and examples of nonconformance found in existing compilers. Since a vendor must revalidate each year, nonconformance decreases.

Validation doesn't say anything about the quality of compiled programs. An Ada Compiler Evaluation Capability is under development, but not yet completed. Some standard Ada benchmarks available are useful in evaluating Ada compilers. You can get information about benchmark software from the Ada Information Clearinghouse.

On each of the PC compilers, I tried out an 8,000-line program that I was developing. I intended to compile the finished program under DEC Ada. The only major features of the language that I didn't use were tasking and low-level features. I reused some abstractions from the Ada repository. I used lots of generic units, including nested generic units, and many packages with private types. I wrote many small subprograms, made heavy use of access (pointer) types, and even used inline pragmas (but only when needed). You'll find the results under the three PC compilers interesting.

To evaluate an Ada compiler, you need to at least consider code quality, programming environment, documentation, and possibly tasking and low-level features. While I couldn't do this in-depth for this review, we will look at some specific topics under each area

(see the sidebar "Evaluating an Ada compiler").

For those of you who aren't very familiar with Ada, I have included a quick, overly simplified explanation of some specific Ada features (see the sidebar "Ada features discussed").

A top-notch compiler from Ada's principal designer

Alsys was founded by Jean Ichbiah, the principal designer of Ada. One of the things I noticed in the documentation for Alsys' compiler and tools was the frequent reference to "highest quality." I guess that shouldn't be too surprising. If you were principal designer of a language used around the world, you'd have a high commitment to quality too. That's not to say that I didn't find anything that I didn't like, but I only found one obscure situation in which one of the tools didn't work.

For those of you familiar with DEC's Ada compiler, I found Alsys' version 3.2 compiler to be of the same quality. I developed my 8,000-line program using Alsys' compiler and had no compiler problems at all. I used the debugger to help me through a couple of coding errors. The only implementation-dependent feature I used was reading the program command line to get the program arguments. When I moved the program to DEC Ada, the only change I needed to make was to use the corresponding DEC function.

I guess the strongest statement I can make about this compiler is that if I were about to start my own company and planned to write PC-based software, I'd get this compiler without any hesitation. Of course, this compiler, with the Developer's Toolset, is more than double the cost of the other PC Ada compilers and their toolsets—but it's worth it!

I tried two versions of the Alsys compiler. I tried the AT version on my AT clone and the Compaq 386 version on my Heathkit H-386. (There is no version for the XT.) Both need DOS 3.0 or later, 640 Kbytes of conventional memory, and 4 Mbytes of extended memory. You need a math coprocessor if you want to use floating-point, or if you want to do fixed-point I/O; I used the 80287 on both computers. You need about 6.5 Mbytes of disk space if you want the Developer's Toolset, which includes Adaprobe, a symbolic debugger and program viewer; AdaXref, a cross-reference generator; AdaReformat, an Ada pretty printer; and AdaMake, a Make utility.

The AT version, which comes with a 4-Mbyte memory board, sells for \$3,595. The 386 version, which doesn't come with a memory board, sells for \$3,095. The 386 version currently only compiles 386 code that will also execute on an AT. The Developer's Toolset costs \$800. The compiler and toolset are available on several other systems such as the PS/2, Sun Workstation, and Macintosh II.

Installation of the compiler and Developer's Toolset were straightforward on my AT clone. On the H-386, I had to make a minor change to the installation batch file. The compiler is really only validated for a Compaq 386 and 100-percent compatibles. You can find out from Alslys if you have to make any changes for other systems.

Programming environment. Most of the Alslys programming environment is provided through AdaWorld, invoked when you type in "Ada." It provides a friendly interface for the Ada program developer. You invoke the Ada compiler and the Developer's Toolset, except for AdaMake, directly from AdaWorld, as you do the program library tools. One of the really nice features of AdaWorld is that you can set defaults for all of the tools.

If you want to tailor AdaWorld's options to the program you're working on, you can set them once and save them in an options file. Since the options file is in your working directory, you can have different options files for different programs, assuming you're using different directories for each program.

For example, before you compile a program, you must create a program library. The Compile and Bind commands always need to know the program library you're using. You can have these two commands default to your newly created program library. If you always want to generate debugging information for AdaProbe, you can set this as a default. You can also set the name of the program you're going to bind as a default. Of course, if you need to override a default for the rest of a session without permanently changing the default, you can do so.

If you only want to perform syntax checking, an option lets you do just that. If you want to do semantic checking as well, but not generate any code, you can do that also.

A really nice help facility built into AdaWorld lets you ask for help on a command, its options, and suboptions. It even allows you to ask for help based on a particular word or phrase. For example "? dependent unit" will yield

all help topics that contain "dependent" or "unit."

I consider the Developer's Toolset a must. AdaProbe is a great window-based debugger that gives you most of the options you need. You can invoke commands from a menu or using keys bound to menu commands. Just as in AdaWorld, you can invoke DOS commands from AdaProbe. Objects are displayed as Ada objects and commands are unit or program-line oriented.

The only capability I found missing was the ability to set a breakpoint on an object to find where it is being modified. I would have liked being able to edit the last command just as I can at DOS command level. Being able to use a mouse with the menus would also be nice. Aside from these and a couple of nits, I found AdaProbe to be a powerful, well-engineered, easy-to-use debugger.

What I found really novel about AdaProbe is its program viewer capability. You can look at program units using different views. For example, you can look at one without comments, with reduced nesting, or both. In the latter case, comments would not appear on the screen and ... would replace a level of nesting. You can control nesting level reduction. You can set the view to see only type, function, and procedure declarations. All sorts of combinations of Ada statements and comments are allowed. You can also have the view apply to the entire program unit or just locally. Program viewing lets you look at the forest, the trees, or the leaves. The choice is yours.

AdaReformat does a really nice job of pretty printing your Ada text. You can control the case of identifiers and reserved words, adjust the amount of indentation, and have the formatting occur on a line-by-line basis, or follow the indentation scheme of the RM. You can even insert a pragma to keep portions of your text from being reformatted. The only feature I found missing was setting the line length. It seems to be set at 72 characters.

AdaXref can be used in one of two ways. You can use it as a dumb Xref tool, in which case it just cross references within a single file. What's really nice about AdaXref is that you can also use it as a smart Xref. In this case, AdaXref uses the program library to cross reference your file, so you get references to other program units. For a language with separate compilation capabilities, this is what you really need!

AdaMake is a nice tool for helping you determine compilation order of Ada program units. It also helps you

Evaluating an Ada compiler

The criteria used in this review are

(1) **Programming environment.** In addition to the compiler, a programming environment should include other tools that can make your Ada programming easier. These tools include special editors, cross-reference generators, pretty printers, debuggers, and syntax checkers. Moreover, tools should assist in maintaining the Ada program library and determining which program units must be (re)compiled (a Make utility). (You'll usually see the program environment for Ada referred to as an APSE.)

(2) **Code quality.** In addition to standard optimization, you will want certain Ada-oriented code improvement.

(a) **Generic units** can be compiled in line or as stubs with runtime descriptors. Each has its place. A user should be able to request either approach on a compilation-unit basis. For multiple instances, a copy of the unit code can be compiled in line for each instance or one copy can be shared using runtime descriptors. The former approach requires more code space than the latter, but is more efficient.

(b) **The Inline pragma** should have reasonable support. This pragma is used to indicate that the body of a particular subprogram should be compiled in line at the point of a call. But pragmas are just advice to the compiler and need not be obeyed.

recompile only those units that need to be recompiled. It's not perfect, as the program documentation admits, but it's good enough.

Code generation. The compiler implements the integer types `Short_Integer`, `Integer`, and `Long_Integer` with ranges that you would expect on a PC. Note that they are true integer types, that is, you can use `Short_Integer` and `Long_Integer` as an array index.

Extensive code optimization and code generation options available with this compiler let you tailor code generation

(c) Subprograms not named in an Inline pragma should be compiled in line when this makes sense. If a local subprogram is only called from one place, the subprogram can usually be compiled in line. This shouldn't require an Inline pragma.

(d) Unused subprogram elimination should take place. If a program unit contains subprograms that are never invoked, they should not be bound in.

(3) Documentation. The documentation should be complete and easy to read. Error messages should make sense. Compilation error messages should relate back to the RM.

(4) Tasking. This should be usefully implemented. In particular, invoking the task scheduler should not be restricted to synchronization points. At a minimum, time-slicing should be provided.

(5) Low-level and implementation-dependent features. These include record representation clauses, address clauses, and machine-code insertions. Record representation clauses allow you to specify the storage representation of record types. Address clauses allow you to place an object at a particular address; place the code of a subprogram, package, or task unit at a particular address; or associate a task entry with a hardware interrupt. For DOS, access to DOS system services should also be provided. The Interface pragma should be provided.

pragma is obeyed.

Alsys Ada provides outstanding support for multiple program libraries. These are useful when you want to break up your development team into subsystem subteams. Each subteam can use its own program library and link to the others. Alslys' implementation provides considerable configuration management control over these libraries. It's easily the best available on the PC and better than much of what you'll see on minis and mainframes.

Programs can be compiled for 286 real mode, protected mode, extended mode, or for an XT. This allows you to break out of the 1-Mbyte limitation of DOS on an AT or 386-based system. A program and its data can thus take full advantage of extended memory. But an individual compilation unit or data object is still constrained to DOS 64-Kbyte segments.

Tasking and low-level features are nicely implemented. For tasking, you can let task switching occur only at synchronization points, or you can specify time slicing at a certain interval.

For low-level programming, a fairly complete DOS system interface is set up as a collection of subprogram calls. This means you don't have to worry about registers. A few system calls are not supported, probably because they might interfere with the Ada runtime system. While I think that should be the user's problem, I found nothing missing that I considered important.

Most of the low-level features allowed by Ada have been implemented. The noticeable exception is support for address clauses. The lack of an address clause is partially offset by a subprogram that you can use to map a task's entry to an interrupt. The Interface pragma is implemented for assembly language and Ada.

Documentation and diagnostics. As an experienced Ada user, I found the documentation very easy to use. It contained all of the information I wanted, and I found its presentation well organized. For example, the material on interfacing with assembly language routines clearly tells you how to do so.

I also found its compile-time diagnostics informative, describing the errors in Ada terms and giving references to section and paragraph numbers of the Ada Reference Manual. For an inexperienced Ada user, the documentation and compile-time diagnostics can help make the transition to Ada considerably smoother.

Compiler: **Reader Service 20**
Toolset: **Reader Service 21**

A surprisingly good, low-cost Ada compiler

Some of you might be surprised to see the Janus Ada compiler reviewed here. I'm surprised too! Until recently, if you had asked me about Janus Ada, I would have said it's just a cheap compiler that implements a not-too-large Ada subset. Boy, would I have been wrong.

This compiler has undergone some radical changes, all for the better. It's a fully validated Ada compiler. It doesn't have all the bells and whistles of other Ada systems, but what it does, it generally does well.

The only problem I found was with generic units in my 8,000-line program. One program unit (String_Scanner from the Abstractions.src file of the Components directory of the Ada Repository) uses a generic package that is itself implemented using a second generic package. The compiler couldn't find the program library information for the second generic unit. While I didn't like this, a fairly simple modification got me around the problem. Eventually, I was able to compile the program.

If you use lots of generic units, like I do, you'll find this annoying too. But you can work around it until the next Janus release.

I used Janus Ada version 4.4 (\$799) and its toolset on my Heathkit H-386. You need at least an XT, with DOS version 3.0 or later and 640 Kbytes of memory. A math coprocessor is recommended, but not required since software floating-point support is provided. You need about 3.2 Mbytes of disk storage.

As of this review, Janus Ada doesn't have a debugger or a program development environment like the other two PC Ada compilers. But I was told that by the time you see this review, they will probably have them. The debugger will cost about \$395 and the program development environment will be about \$250. If the tools are as good as the compiler (sans the problem with generics), competition between the PC Ada vendors may heat up at the low-cost end.

If I were going to buy a low-cost Ada compiler for the PC, I'd probably pick Janus Ada. There are a few things about it I don't like, but most are minor problems.

Programming environment. Right now, the programming environment of Janus Ada is somewhat limited. But it does have a nice collection of little extras. It has a Make function that

to your needs. In fact, you can avoid increased compilation time until you're ready to bind a production version of your program.

Code generation options allow you to obtain smaller or faster programs. For example, you can turn off all runtime checking, or all but stack overflow checking. You can control whether unused subprogram elimination takes place. You can indicate on a compilation unit basis whether or not generic units are compiled in line. You can control whether or not subprograms that can be generated in line actually are, as well as whether or not the Inline

works quite well, although I think the interface could be simplified. It has a syntax checker and a pretty printer. A profiler analyzes your programs. To use it, you compile your main program with the profiler package, make a couple of initialization calls, and you're off. Another useful tool will search a file looking for a pattern described by a powerful regular expression language.

The real surprise is the assembler. It's a full 8086 assembler, with some of Ada's syntactic sugaring built in. It supports an Ada-like If statement, expressions, identifiers, comments, and pragmas.

In particular, you can write a normal Ada specification with type, object, exception, and subprogram declarations. The body of the package specification can be implemented using the assembler, and you can use these declarations within the body. You can use declared objects reasonably well, and you can use dot notation for record fields.

The assembler's limitations will not affect your normal use of it. When you do use the assembler, it's nice to know you don't have to stray too far away from the Ada view of things. By the way, Janus Ada also provides a disassembler for programs produced by the compiler. And of course, it uses the notation of its assembler.

I strongly disliked Janus Ada's assumptions about file names and suffixes. For some of the tools to work properly, the compiler assumes the file name to be the first eight characters of the program unit. I think that's absurd! As a user, I want my file names to tell me something about the file contents. I think this should be changed as soon as possible. Also, the Make program assumes specifications have the suffix lib and bodies have the suffix pkg. You have the option of adding one additional suffix for each. Again, I think that's absurd!

Code generation. Janus Ada only supports Integer as 16 bits. Rather than having a Short_Integer for an 8-bit Integer, it has Byte instead. What I find really strange is that it has a 32-bit Long_Integer, but it's not a full-fledged integer type. You can perform arithmetic on this type, but you can't use it as an array index.

Although it took second place to the Alsys compiler, Janus Ada did a good job of code optimization. It does support unused program elimination. However, it doesn't support the Inline pragma and doesn't compile other subprograms in line. It also doesn't compile generic instances in line.

Programs and data are limited to conventional memory, so any extended memory you have can't be used. Janus Ada also restricts the size of a compilation unit or data object to a 64-Kbyte DOS segment.

Tasking isn't particularly well supported, since the task scheduler can only be invoked at synchronization points. Address clauses are not supported, so you can't map a task entry to an interrupt.

This compiler has the weakest support for DOS system calls. Some of the program control services are provided directly as subprograms, but the rest of the system calls must be accomplished using a half dozen subprograms. These latter subprograms require a record of register values to be passed as in/out parameters. The other low-level support is reasonably good, but pragma Interface isn't supported.

I'm not sure how to describe Janus Ada's program library. It's not an entity of its own as in the other PC Ada compilers. Instead, it's a collection of

files generated in the source directory. Whether it can be used to support multiple libraries, I couldn't say for sure, but the documentation doesn't mention it one way or the other. Anyway, library support is much weaker than the other PC Ada compilers.

Documentation and diagnostics. The documentation for Janus Ada is outstanding, with only a few exceptions. The manual is organized along the lines of the RM. Explanations of Ada features and their implementation in Janus Ada are clearly discussed. While this could annoy someone well versed in Ada, it's fantastic for a newcomer. A 175-page Ada tutorial provided with the compiler, although far from enough to teach you Ada, can make you aware of some basic concepts.

The documentation fell down a little in describing compiler and linker options. A little bit of technical editing would help.

Reader Service 22

Ada features discussed

Among Ada's predefined types are Integer and possibly Short_Integer and Long_Integer. On the PC, you might expect Integer to use 16 bits, Short_Integer to use 8 bits, and Long_Integer to use 32 bits.

An Ada package is a collection of related definitions: constants, types, objects, subprograms, etc. Typically, a package provides a single abstraction. The specification of the package presents the view of the abstraction allowed users of the package. The body of the package contains most of the implementation details of the abstraction and hides them from the abstraction user. To further hide details of the implementation, types in the specification can be defined as private types. Essentially, the only operations allowed on objects in a private type are those provided for it by the package specification.

For example, a package providing an Integer priority queue abstraction (ordered by "<") can define the Queue_Type as a private type. The package can provide subprograms for adding to the queue, removing from the queue, etc. Users of the queue can't take advantage of the implementation details of the queue.

Ada also allows generic subprograms and packages. These are parameterized versions, where the parameters might include types and subprograms. The actual types and subprograms are supplied during compilation to form an instance of the unit.

For example, a generic Max function might have the value type and an ordering function as its generic parameters. To create an instance of an Integer Max, just supply Integer as the actual type parameter and "<" as the ordering subprogram. If Person is a user-defined type and Larger_SSN orders two Person objects by Social Security number, then these can be used to create a Max function for Person objects. A generic version of the priority queue that takes the type of the objects to be queued and a priority function for these objects can be used to create a queue instance for any type object, as long as a priority function exists for objects in the type.

Packages, especially generic ones, encourage the use of a large number of small subprograms. Since many of these are compiled separately from the program units that use them, subprogram call overhead can

A low-cost, moderate-quality Ada compiler

Meridian's AdaVantage was the first low-cost validated Ada compiler for the PC. Version 2.1, which I reviewed, is a complete implementation. In addition to the compiler, it has a debugger and an interactive development environment.

I very much want to like this product, but some things about it bother me. When you look at the complete package for AdaVantage, it seems quite impressive. But I noticed a lack of attention to detail. Part of this shows up in the documentation binder. It's much too crowded. At least two or more binders would make the documentation easier to deal with. Then there's the index that doesn't look complete. And the typography used seems to blur the distinction between examples, text, and section separations.

If that were all, I'd feel better about this product. But I saw the lack of

attention to detail in the compiler as well, when I tried to compile my 8,000-line program. I didn't get very far because of problems with generics. One of the generic packages (Lists from the Abstractions.src file of the Components directory of the Ada Repository) contains a generic subprogram, CopyDeep, which has a generic function parameter named Copy. The compiler had all sorts of problems when the body of CopyDeep referenced Copy. Moreover, the compilation error messages were misleading. All of this seems to indicate that generics haven't been implemented very well.

The compiler had problems with generics in version 2.0. It used to hang the system when it encountered some of them. While it doesn't hang the system anymore, it still makes generics not very usable. I was originally going to review version 2.0, but for various reasons I didn't get around to it. I did tell Meridian about the problems I was having with generics and I did send them all of the Ada repository that I had collected,

pragma is used for a subprogram or the subprograms provided by a package, then this changes. If the body is compiled in line and the body changes, then the unit containing the in-line code must be recompiled. This increases recompilation dependencies.

Ada tasks are program units that can execute concurrently. A task specification generally contains entries through which other tasks can communicate with it. When two tasks want to communicate, they must do so synchronously, through a rendezvous. For a rendezvous, one task, when it's ready to do so, calls the entry of a second task. The second task, when it's ready to do so, accepts the call. The rendezvous, task activation, and task completion are synchronization points. A task can serve as a hardware interrupt handler by using an address clause to map one of its entries to a hardware interrupt vector.

The Interface pragma allows Ada programs to call subprograms written in other languages. Whatever language interfaces this pragma supports, any external subprogram that follows one of these conventions can be invoked from an Ada program.

including the abstractions. I would have expected this to be fixed in the 2.1 release.

With a little more attention to detail, I think AdaVantage could be a good low-cost entry into Ada on the PC, especially if you have a limited budget and don't need extended memory. I'd like to see AdaVantage after another major release.

I used AdaVantage (\$795) and its toolset on my Heathkit H-386. You need at least an XT, with DOS version 3.0 or later and 640 Kbytes of memory. A math coprocessor is recommended, but not required since software floating-point support is provided. The full product line, with utility (\$50) and environment (\$50) packages, debugger (\$495), development environment (\$170), and optimizer (\$200), requires about 2.5 Mbytes of disk space. If you're going to buy AdaVantage, you should get the whole product line.

Programming environment. The Ada Developer Interface (ADI) is a screen-oriented program that integrates the compiler, linker, Ada template editor, pretty printer, and debugger. ADI is icon-based, but I found the icons distracting. It's not that I don't like icons, it's just that I found the icons to be more flash than substance. I also found the special symbols lacking any intuitive meaning, so you have to remember what they mean. That's fine if this is the only program you're going to use, but most of us use many programs. Also, my H-386 has VGA color, but ADI gave me a dull monochrome display.

Anyway, the idea is sound. You have icons for functions like editing, compiling, formatting, etc. Moreover, the icons exist on a program-unit basis, for both specification and body. You can draw a tree of a unit's dependent units and see their status. With more attention to the presentation, ADI would be a really good tool.

Even with problems, ADI has some nice features, including the Ada Template editor. You can invoke an EMACS-like editor and bring up Ada templates. For example, you can bring up the template for an If statement and fill it in without having to type in "if," "then," and "end if." This can save you lots of time when you're writing an Ada program. While AdaVantage doesn't have a syntax checker, if you use the template editor, you won't have much call for one.

I did find one really annoying thing about ADI. To invoke the function associated with an icon, you position the cursor over the icon and press the

sometimes be a problem with frequently called subprograms. The Inline pragma can remove this overhead.

In contrast to the often-used subprograms of an abstraction are subprograms not used in an application. Ada encourages reuse of abstractions from one application to another, so when you write an abstraction, it is usually complete. But this means that not all applications will use all of the subprograms. For example, a pattern-matching abstraction might contain Match and No_Match subprograms. If an application only uses Match, it shouldn't have to include code for No_Match. Unused subprogram elimination would not bind in No_Match.

Ada's separate compilation capability requires some notion of a program library. This library contains all of the information necessary to ensure that a separately compiled specification is used correctly by other units. Generally, if a specification changes, after it is recompiled any units using that specification must also be recompiled. Normally, the body of a unit can be recompiled without requiring units that depend on it to be recompiled. If an Inline

Enter key. That's fine, but it's easy to forget and start typing in a function name instead. I did that with the editor. Unfortunately, the letter "e" by itself is also the exit command. I did not appreciate being thrown out of the program because of a minor mistake. Attention to detail?

AdaVantage has a primitive Help command that I found useful. While you'd never mistake it for an advanced capability, it did the job for me and it's more than you get with many software products. It was nice not to have to get out the user manual just to find out what switches I needed to specify.

The debugger is really powerful and has lots of added features that, while not necessary, can be useful. For example, a single command will display a subprogram's local variables. Of course, they are displayed as Ada objects. By the way, trapping on writes to an object is supported. In general, the debugger is more the type that you would expect on a mainframe.

Code generation. AdaVantage supports Integer using 16 bits and Long_Integer using 32 bits. For some reason that I can't understand, it uses Byte_Integer rather than Short_Integer for 8-bit integers. Anyway, these are true Ada integer types.

The AdaVantage optimizer did a fairly good job of reducing program size, but the programs it produced consistently had longer execution times than those produced by the other PC compilers. The Inline pragma hasn't been implemented and there's no indication that in-line generation of other subprograms has been implemented. Instances of generic units are always generated in line. Also, the optimizer provides unused subprogram elimination.

Tasking support is limited to invoking the task scheduler at synchronization points. This limits the usefulness of tasking under AdaVantage.

AdaVantage's multiple library support is very nice. You can have different teams sharing libraries. However, it lacks configuration management support for the libraries.

Of the three PC compilers, AdaVantage seems to have the best low-level implementation. It has two optional packages—Utilities and DOS Environment. You don't need them unless you're going to interface with the DOS environment. But you need the DOS Environment package even if you just want to read the command line. The other PC compilers provide the equivalent features of these packages at no extra charge.

Utilities provides you with bit-level operations for word and double-word operands. These operands are provided as Integer and Long_Integer, although unsigned integers would have been more appropriate. The standard trigonometric functions are also provided. Variable-length strings are provided through an implementation of the Text_Handler package whose specification appears in the Ada Reference Manual.

DOS Environment provides a logical interface for all DOS interfaces. It also contains packages that provide frequently used capabilities. For example, Box allows you to draw boxes of various sizes. Cursor provides you with complete control over the cursor position and size. Program_Control allows control over executing programs. TTY provides terminal and keyboard control, while Video provides support for monochrome, color graphics, and extended graphics adapter cards. And you don't need to deal with registers.

AdaVantage completely supports address clauses and provides machine-code insertion. This latter feature allows you to write small sequences of 8086 instructions within an Ada subprogram. This is an advantage over an assembler, in that the compiler can optimize these instructions as well. It also supports the Interface pragma for assembly language and Meridian C.

Just as for the other PC Ada compilers, the DOS limits of 64 Kbytes for a single compilation unit or data object apply. Also, code and data are limited to conventional memory, so even if you have extended memory, you can't take advantage of it.

Documentation and diagnostics. As I already mentioned, AdaVantage documentation leaves something to be desired. I found the ADI documentation especially poor. The tutorial was so limited that it didn't show off much of ADI's capability at all. By the way, if you are familiar with version 2.0, compilation error messages now reference RM sections when appropriate.

Reader Service 23

Tools that make Ada more attractive to use

Two tools that can make your move to Ada a little bit easier are an on-line reference manual and a Pascal-to-Ada translator. While these tools execute on a PC, you can take advantage of them

even if you do your work on some other system.

Mastering the Ada Reference Manual. When you're trying to figure out how to do something in Ada, or why something isn't legal, or if something is legal, what do you do? Some people pick up their favorite Ada text and try to find the answer there. But most books don't have all the details. People who have been using Ada for years may pick up the RM, but if you're just getting started, it's hard to find your way around—and the RM can be intimidating. Even for people who are comfortable using the RM, moving from place to place in the RM and keeping track of the sections visited can take a lot of effort. No matter what your needs, AdaQuery can be a big help.

AdaQuery is available from Alslys for \$200. Its modest system requirements are a PC, 256 Kbytes of memory, a monochrome or color display, two diskette drives, and DOS 2.0 or higher. If you have a hard disk, you need about 640 Kbytes of storage space. I used my H-386, with 640 Kbytes of conventional memory, color monitor, and a hard disk.

Common to all commands are the active and alternate windows. You can split the screen between the two, put the results of a command in the other window, and make it the active window. Of course, if you want to put the results in the same window, you can. Several commands let you move within and between windows with very little effort.

AdaQuery deals with parts of the RM. These are the Table of Contents (TOC); any chapter, section, or subsection; any annex or appendix, and the Index. It also includes a list of technical terms.

When you invoke AdaQuery, only the active window is displayed. It contains the TOC. You can position the cursor at any chapter or section number, or on any annex or appendix letter. Hit a function key and you're at the corresponding part.

In any part of the manual, certain technical terms are of interest. You can move between them using the arrow keys. You can hit a key and bring up the index entry for the term. Once you're at the index entry, positioning the cursor at a particular reference number or letter and hitting a key gets you to the part of the manual containing that reference. If you position the cursor on a technical term, you can call up its definition with a keystroke.

A single key will bring up the TOC or list of technical terms. A single key will also tell you what part of the manual

you're in.

There's a really nice matching facility. You can give up to five words, or parts of words, and ask for matches on all or some of the words. For example, the words "type" and "declaration" can be used to find all matches containing both words, either word, both words and others, or either word and others.

Finally, you'll really like the history facility. Whenever you reference a technical term, the term and the part number you were using with it—together called a reference item—are placed in a history file. You can call up this history file at any time, position the cursor at a reference item, and bring up the part again. You can move the reference items about, delete those that you no longer want, and even set up a chain of reference items that you can follow, bypassing ones that you're not interested in at the moment. You can save the history file and load it later.

I really like AdaQuery, and at \$200 it's a real bargain!

Reader Service 24

Passing from Pascal to Ada

If you're going to use Ada on the PC, there's a good chance that you have a legacy of Pascal software. What are you supposed to do with this, especially if you need to deliver source to a customer who insists on Ada only?

The best thing would be to redesign your Pascal software in Ada. That's how you're going to get the best Ada code. But you may not have the time or money to do this from scratch.

Pascal is close enough to Ada that an automated translation might produce reasonably good Ada code. Then you have the option of keeping it as is, or massaging it into even better Ada code. But first you need a tool that's going to do a pretty good translation.

PasTran, from the makers of Janus Ada, is what you need. It requires a minimal PC system: 256 Kbytes of memory, two 5¼-inch floppy disk drives, a monitor, and DOS 2.0 or later. It's got a hefty price of \$2,500, but it's definitely worth it.

PasTran accepts ANSI/IEEE standard Pascal. It has three passes. The first pass checks syntax and tries to fix errors that it finds. You can have it do this automatically or have it ask you first. In the second pass it performs a semantic analysis of the first pass results. Finally, the third pass produces

the Ada program using the results of the second pass.

Now, what if your programs aren't in ANSI/IEEE standard Pascal? Well, PasTran tries its best to go on even if it finds errors. So you can submit your Turbo Pascal, or whatever, programs and get most of the translation done for you. Of course, you probably should make some modifications before attempting translation.

Actually, PasTran recognizes a few Pascal extensions. For example, identifiers can have underscores, constants can be full expressions, etc. So the translation may be easier for some Pascal dialects.

I'd like to see PasTran come with support for several popular Pascal variants, such as Turbo Pascal, Microsoft Pascal, DEC Pascal, etc. That would make this great tool even easier to use.

A few things PasTran won't do. For example, it won't handle buffer variables. It treats them as undefined identifiers. But things that you expect it to do, it does. For example, a Pascal value parameter, A, is translated to an Ada in-parameter named Q_Value_A and a local variable, A, initialized to Q_Value_A.

I tried PasTran on some large public-domain Pascal files and it worked great. I had to make a few changes to the Pascal and Ada files because the program was in DEC Pascal, but it only took a few hours. Since I didn't know the program, I figured that a straight translation would take about a week or two, if I tried it manually. That sold me!

Anyway, if you know that you have to convert Pascal code to Ada code and you've been dreading it, you know what to do—get PasTran.

Reader Service 25

Summing up

After looking at the three Ada PC compilers, it's clear that Ada is well supported on the PC. You have a choice of cost and quality—whatever meets your needs. Many tools available can help make your move to Ada a little bit easier. The two I looked at just scratch the surface, but will be of use to many of you.

If you're developing software just for the PC, or if you're developing software that needs to run on minis and mainframes as well, there's no longer any reason to avoid using Ada. Ada is for real on the PC. With 80386 and Unix/Xenix-386 support likely by the

end of the year, the picture's only going to get brighter.

Product notes

- Korenthal Associates (230 W. 13th St., New York, NY 10011) has developed a program called 4Print (version 2.20) to output plain ASCII (or Wordstar) files to any HP printer equipped with a small, landscape font. Such a printer can pack two 66-line pages, plus title, onto one side of one physical page (in landscape orientation). By using two-sided printing (also supported), four pages of listing can be packed onto one piece of paper. This shareware program costs \$25 for registration, plus another \$10 if you get it on disk from them.

- CF4000 is a low-cost schematic design package tailored for the Sun workstation. The system includes a schematic and symbol editor, a symbol library, plotting and parts-list programs, a generic ASCII netlist, and options for the creation of custom netlists with back annotation. Priced at \$995, the system is available from Phase Three Logic, PO Box 985, Hillsboro, OR 97123, phone (503) 640-2422.

- Aewindos is a windowing package for C programmers from Aesoft Corp., 2570 Woodstock Place, Boulder, CO 80303, phone (303) 499-7332. The software library works with Turbo C, Microsoft C and Quick C, Lattice C, and Power C. Tools that come with the package include RAM-resident on-line documentation, a RAM-resident snapshot facility, and a window editor.

- Looking for quick-turnaround ASIC production services? A system from Lasarray Corp. combines an FET-level silicon compiler design workstation with a direct-write laser fabrication tool. The 750-square-foot, self-contained fabrication facility is described as a "silicon printer" because when approved design and gate-array base wafers go in, finished devices come out the next day. For more information, contact Lasarray at 13845B Alton Parkway, Irvine, CA 92718, phone (714) 581-0889.

NEW PRODUCTS

New IBM series based on ESA/370

IBM has announced the Enterprise System/3090 Model 600S, which the company calls a general-purpose processor. The Model 600S falls at the upper end of a 10-model series of main-frame computers using IBM's Enterprise Systems Architecture/370, the MVS/ESA and VM/XA operating systems, and data management software.

Prices for the ES/3090 S models range from \$985,000 for the 120S to \$12.4 million for the 600S. The 120S and 150S are scheduled for availability in September; the models 170S and up are scheduled for the fourth quarter of 1988. Upgrades will be available in the first half of 1989.

Reader Service 30

General-purpose RISComputer delivers 20 MIPS

MIPS Computer Systems has announced the M/2000 RISComputer based on the company's R3000 chip set. The company claims that the new general-purpose computer delivers sustained system performance of 20 MIPS.

The base-configuration M/2000 features a 25-MHz R3000 RISC processor, an R3010 floating-point processor, 128 Kbytes of instruction and data caches, 32 Mbytes of ECC main memory, a 714-Mbyte ESMD disk drive, and one Ethernet controller with support for up to three more. The M/2000 is an 18-slot, VME-based rackmount package.

Standard software includes the company's RISC/os port of the Unix oper-

ating system, an optimizing C compiler with symbolic source-level debugger, and support for Ethernet, TCP/IP, and NFS.

Options include up to 128 Mbytes total of main memory, up to eight ESMD disk controllers supporting up to 3 Gbytes in a single bay or 17 Gbytes in a triple bay, a magnetic tape drive, and unlimited serial I/O ports.

Optional software includes Fortran, Pascal, Ada, PL/1, and Cobol.

Prices range from \$100,000 to \$150,000 depending on the configuration.

Reader Service 32

GoldWorks AXLE trains AI developers

Gold Hill Computers offers a software package called GoldWorks AXLE (for Albathion Expert Learning Environment) that illustrates basic and advanced techniques for expert system development. The software was developed by Albathion Software and is marketed and distributed by Gold Hill.

According to Gold Hill, the program goes step-by-step through the development cycle, using applications for diagnostic and planning expert systems as examples and providing source code for these examples.

AXLE demonstrates four expert systems: manufacturing and aerospace diagnostics applications and industrial and oil-exploration planning applications. The examples reportedly demonstrate methods for notifying operators of mechanical problems (diagnostics) and for decision making under constraints (planning).

The program includes an interface

called the Navigator, an interface called the Browser, the Graphics Toolkit for building object-based graphical interfaces to a GoldWorks knowledge base, and development modules.

AXLE can be used by itself or with GoldWorks, the company's expert system building tool. Alone, it requires 5 Mbytes of extended memory and 7.5 Mbytes of free disk space. Used with GoldWorks, it requires 2 Mbytes of extended memory and 3.5 Mbytes of hard disk space in addition to GoldWorks requirements. It also requires 512 Kbytes of base memory, a CGA display adapter, and a monitor.

The software runs on an IBM PC AT or compatible, IBM PS/2, Compaq 386 or compatible, or 386 HummingBoard. It comes with four manuals.

GoldWorks AXLE costs \$1,995.

Reader Service 31

IBM adds new RT systems

IBM has announced three new RT systems. The 32-bit workstations use a RISC processor. Model 130 is a desktop unit, while models 135 and B35 are floor-standing systems. According to the company, they can be used alone, in a multiuser network, or connected to a host.

Models 130 and 135 require an RT-supported display and adapter as a minimum configuration. Model B35 must be attached to an IBM 5080 Graphics System.

All three models require the AIX/RT Operating System Version 2.2 or equivalent. All come with 16 Mbytes of CMOS memory, an enhanced version of the IBM Advanced Floating Point accelerator, and 114 Mbytes of fixed disk storage.

Model 130 costs \$23,220, model 135 costs \$30,595, and model B35 costs \$32,165.

Reader Service 33

Prime expands 50 Series with midrange office model, entry-level computer-room model

Prime Computer has added the office-installable midrange 4450 system and the 6150 entry-level computer-room system to the 50 Series of super-minicomputers. Both run the Primos operating system. The central processors incorporate emitter-coupled logic components and VLSI designs, according to the company.

The 4450 system supports up to 254 users and up to 3 Gbytes of data. Connection to an uninterruptible power supply is standard. The system features a diagnostic processor, 32-bit word lengths, 32 Mbytes of main memory, an 857-ns memory cycle time, 64 Kbytes of cache memory, a 63-ns cache cycle time, a 46-ns effective memory access time, 512 Mbytes of virtual address space, and an I/O bandwidth of 9.5 Mbytes per second.



Prime Computer claims processing power up to 5.8 MIPS for its 4450 system.

Standard 4450 systems begin at \$290,700 and range to \$353,800.

The 6150 incorporates a five-stage instruction pipeline and an associative write buffer. It supports up to 64 Mbytes of memory and 512 user terminals. The system has floating-point instructions built into the hardware. Other features include 32 Kbytes of cache memory, a 34-ns effective memory access time, 512 Mbytes of virtual address space, maximum disk storage of 24.6 Gbytes, support for up to eight tape devices, and support for up to 960 user processes.

The 6150 system begins at \$443,400 and ranges to \$554,300.

4450: Reader Service 34
6150: Reader Service 35



Prime Computer claims processing power up to 8.5 MIPS for its 6150 system.

Neurocomputing coprocessor fits Sun-3 slots

Hecht-Nielsen Neurocomputers has announced the Anza Plus/VME, a neurocomputing coprocessor board for Sun-3 workstations. The board installs in any Sun-3 VME slot, according to the company. It comes with 10 Mbytes of RAM, plus software for the integration of neurocomputing into C programs.

The coprocessor can reputedly implement neural networks with any combination of up to 2.5 million processing elements and interconnections. It updates the network at a peak rate of 10 million interconnects per second (1.5

million IPS in learning mode).

The software is Release 2.1 of the company's neural network software system, operating under the Sun Unix operating system. It includes three subsystems: the Neurosoft package of five neural network paradigms, the User Interface Subroutine Library to link neural networks and C programs, and the NetSet user interface.

The Anza Plus/VME costs \$25,000 in the US and \$31,000 outside the US.

Reader Service 36

TeleSoft releases new Ada

TeleSoft has released Sun Ada 1.3, an updated Ada development environment for Sun-3 Unix-based workstations. Sun Microsystems distributes Sun Ada 1.3.

Sun Ada 1.3 is based on TeleGen2. New features include math libraries, the Interface pragma to Fortran, an extended input list capability, and an enhanced source-level debugger.

Sun Ada includes an Ada compiler and a set of development and productivity tools. It has been validated by the DoD's Ada Compiler Validation Capability version 1.9.

TeleSoft has also announced that Sun Ada now supports X Windows, a graphics package developed by MIT. Sun Ada 1.3 costs \$7,500.

Reader Service 37

N-Net helps implement neural net technology

AI Ware offers N-Net, based on the Functional Link Net architecture. N-Net reputedly assists implementation of neural net technology across a broad range of application areas.

According to the company, the Functional Link design provides an integrated environment for supervised learning, unsupervised learning, and associative recall within the same command and data structure, and within the same net. Users need not reconfigure the network in going from one type of learning algorithm to another.

N-Net also features a proprietary "front end" that reportedly massages training samples before they are learned, for increased accuracy and an improved learning rate.

N-Net requires an IBM PC, XT, AT, PS/2, or compatible with 512 Kbytes of RAM, a hard disk drive, and DOS 2.0 or higher. It requires Microsoft C for the programming interface. For the Windows interface it requires a color monitor; CGA, EGA, or VGA graphics card; mouse; and Microsoft Windows.

The company suggests adding the optional N-Net Learning Card to increase learning rates. The card costs \$3,000.

N-Net 210 costs \$695. A demonstration kit costs \$49.

Reader Service 38

Wyse 386-based systems handle many users

Wyse Technology has added two multiuser computers to its line of systems based on the Intel 80386 chip. According to the company, the WY-3216-150T supports up to 12 users and incorporates a 150-Mbyte ESDI disk drive with a 16-ms access time, while the WY-3216-85T supports up to eight users and has an 85-Mbyte drive with a 23-ms access time.

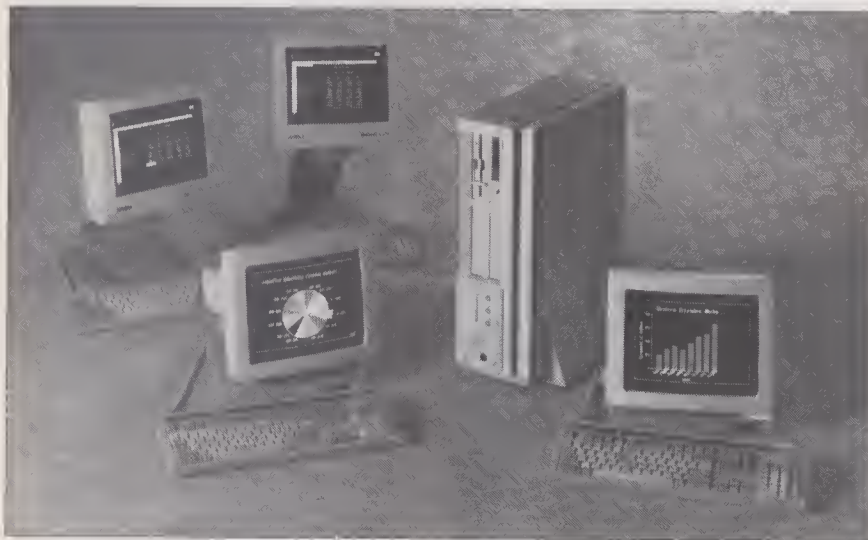
Both models include a 60-Mbyte, 1/4-inch cartridge tape drive and a Sytos

tape utility that allows the tape system to be used in MS-DOS environments. Each system also includes 1 Mbyte of 32-bit memory and a 1.2-Mbyte floppy disk drive.

Wyse resellers can add the Wyse/SCO Xenix multiuser operating system.

The WY-3216-150T costs \$8,599. The WY-3216-85T costs \$7,999.

Reader Service 39



Wyse Technology's 386-based Model 3216-85T comes with an 85-Mbyte drive, while the Model 3216-150T comes with a 150-Mbyte drive.

Toolkit accesses hypertext graphics

KnowledgePro, a knowledge processor from Knowledge Garden, can access graphic images from hypertext using the company's new Graphics Toolkit. The add-in product reportedly allows KnowledgePro users to link graphic images within the system, define hot spots that link to a sequence of program commands, point at one part of a picture and see a message or another picture with further hot spots, run another knowledge base, perform calculations, execute a set of If . . . Then rules, run an external program, and so forth.

The Graphics Toolkit requires an IBM PC, XT, AT, or PS/2 system with EGA resolution, 512 Kbytes of memory, and a hard disk drive. A mouse is recommended, but not required.

The Graphics Toolkit costs \$89. The KnowledgePro development environment costs \$495.

Reader Service 40

Software helps shape expert systems

Perceptics offers Knowledge Shaper for VAX/VMS and Sun/SunOS computer systems. According to the company, the software provides an environment for knowledge acquisition and manipulation, a tool for validation and verification of knowledge bases, and an environment for the design and development of expert systems.

Knowledge Shaper accepts expert knowledge as production rules or decision trees. It does not restrict the definition of objects and their attributes. The company claims that the software produces optimal decision-tree representations of the decision-making process and optimal forms of the solution. Production of high-level source code reportedly provides portability of the resulting expert system.

Knowledge Shaper costs \$4,995.

Reader Service 41

IBM puts out new DOS

IBM has announced DOS 4.0. The new version of DOS reportedly supports expanded memory and very large files. According to the company, it contains more than 30 new and enhanced functions. It also comes with a program warranty and IBM service.

DOS 4.0 will be available in 14 national languages. It runs on all versions of the PS/2, plus the IBM PC, PC XT, PC AT, Portable PC, and PC Convertible with a minimum of 256 Kbytes of memory.

DOS 4.0 has a one-time license charge of \$150. Upgrades cost \$95 until April 28, 1989.

IBM has also announced the availability of OS/2 Extended Edition 1.0. It has the features of OS/2 Standard Edition 1.0 plus a relational database manager and a communications manager. The one-time license charge is \$795. There will be no charge for an upgrade to Extended Edition 1.1 (scheduled for November).

DOS 4.0: Reader Service 42
OS/2: Reader Service 43

ASIC design tool ties in to cell library

NCR Microelectronics Division has announced NCR Design Synthesis. The design synthesis tool for cell-based application-specific ICs reportedly uses high-level behavioral language descriptions rather than circuit schematics. It also integrates an ASIC design system and cell library.

The product belongs to NCR's Knowledge-Based Engineering Environment, KE². It is coupled with NCR's ASIC cell libraries, including both 2-micron and 1.5-micron digital and analog cells, supercells, and compiled functions.

The Design Synthesis system employs expert rules for ASIC designers to specify and simulate their ASIC architecture in conceptual terms. It runs on Mentor Graphics workstations.

NCR will offer Design Synthesis both as a design service and as software for Mentor Graphics workstations. The service will cost approximately \$5,000 for a medium-complexity block, or \$51,500 for a license.

NCR has also announced that it has added fault grading capabilities to its ASIC design services. The base price for probabilistic fault grading is \$2,000.

Reader Service 44

Exsys system works with DOS, Unix, VMS

Exsys offers Exsys Professional for development of advanced expert systems. The product was designed around the original Exsys Expert System Development Shell. Professional helps generate applications compatible across MS-DOS, Unix, and VAX/VMS environments, according to the company.

Features include a command language, access to data in dBase III files, a rule compiler, custom screen capabilities, two new confidence modes, multilevel password protection, enhanced math capabilities, and so forth.

The command language reportedly provides control mechanisms for input of data, execution of rules, looping, and display of results. According to the company, the command language resembles a batch procedural language to control the flow of the expert system, while the rules provide the logic, inference, and chaining capabilities.

Professional is also available in Japanese for MS-DOS and Unix computers. Exsys Professional costs \$795.

Reader Service 45

Unix gets 4GL/DBMS in multi-part Apogee

Binary Research has announced what it calls a fourth-generation language and database management system, Apogee QP/UX. Programmers can reportedly write applications on an MS-DOS PC and port them to Unix operating systems.

Apogee features an editor with full-screen editing of ASCII files, creating and painting windows, and syntax checking. An interpreter with 140 commands contains utilities to optimize program code.

The software can also translate programs into C using its own C translator.

Apogee SQL/RG is a structured query language and report generator. Apogee Windows, a windowing pro-

gram, comes bundled with other Apogee products.

Prices for Apogee products range from \$395 to \$4,500 according to the environment and elements included. A bundled package including the editor, interpreter, C translator, and SQL/RG costs \$4,500 for Windows and non-Windows versions alike. Apogee QP/UX costs \$2,500 in the Unix environment. Apogee SQL/RG costs \$1,000 in the Unix environment, with or without the Windows package.

The Apogee products come with a terminal and keyboard setup program.

Reader Service 46

KBMS targets IBM mainframe computers

AICorp has developed KBMS, a knowledge base management system for IBM mainframe computers, in conjunction with a consortium of clients. According to the company, KBMS incorporates four major design elements: object-oriented programming, a multiuser architecture for transaction processing, a data access system, and a knowledge manipulation system.

Object-oriented programming supposedly allows for fewer rules than otherwise, with application logic being independent of data manipulation.

The multiuser architecture includes multithreaded processing for recovery and integrity of the application. It also features a central server architecture for concurrent access from different subsystems, code written in C, and RETE network technology indexing the knowledge base for direct rule access.

The program's Data Flow System accepts input from various sources and

automatically handles the external data access, generating SQL for relational database management systems. According to the company, DFS accesses data only when all conditions are satisfied and offloads processing to the DBMS when appropriate.

KBMS uses SQL to communicate with procedural application programs and user routines. It communicates multiple records at a time. It also allows programmers to use various SQL functions.

KBMS supports the VMX/XA, MVS, and VM operating systems; CICS, TSO, IMS/DC, and CMS teleprocessing monitors; and DB2, SQL/DS, and other database management systems.

KBMS costs from \$90,000 to \$225,000 depending on the options and database interfaces chosen.

Reader Service 47

Fortran compiler runs on transputer boards

3L Ltd. has announced the Parallel Fortran compiler, which runs on PC-hosted transputer boards like the Inmos B004. According to the company, transputer programs (including the compiler) are run directly from the DOS prompt and can transparently access DOS files and interrupts.

The software reportedly supports full ANSI Fortran 77 with runtime library extensions for intertask communication and dynamic process creation. Configuration software supplied with the compiler makes separately compiled Fortran tasks execute in parallel.

Applications can be developed on a single transputer, then run on a multitransputer system for higher performance, according to the company.

The Parallel Fortran compiler costs \$750.

Reader Service 48

LANlords connect Cray, DEC, IBM systems

Computer Network Technology has announced the LANlord 8000 series of interprocessor networking gateway products and the LANlord 6000 series of networking products for users of Cray Research station software.

The LANlord 8000 series implements TCP/IP communications protocols to support communications between Cray supercomputers and IBM and DEC computers, as well as any computer based on Ethernet networks.

The initial product, the model 8023 interprocessor gateway, connects up to four Ethernet links to the I/O channel on Cray supercomputers. The 8023 interprocessor node reportedly supports TCP/IP protocols provided by Cray as part of the Unicos operating system. The model 8023 handles peer-to-peer communications between the Cray supercomputer and any Ethernet-based computer also implementing TCP/IP protocols.

The LANlord 6000 series supports Cray station communication software.

The 8023 configured to support two Ethernets to one Cray channel costs \$38,000. Support for two additional Ethernet networks costs \$6,000.

LANlord 6000 product prices start at \$38,000.

8023: Reader Service 49

6000: Reader Service 50

Alslys plans Ada compilers for Inmos transputer

Inmos and Alslys have agreed to develop validated Ada compilers for the Inmos transputer, a VLSI microprocessor with an enhanced RISC instruction set. The compiler design reportedly follows a design study carried out by Alslys, Inmos, and Meiko.

Initial products, scheduled for July 1989, will include a cross compiler run-

ning on a DEC VAX under VMS and a host compiler running on an IBM PC card under MS-DOS using the IMS T800. Code will be targeted to the family of 16-bit and 32-bit transputers, including the IMS T800 floating-point transputer.

Reader Service 51

Software powers math on workstations

MathSoft has announced MathStation, a software package for workstations. According to the company, MathStation helps users develop mathematical applications and models and perform numerical analysis on engineering workstations.

The company claims that users can customize the calculation capability, graphical display, and user interface. Users can also define operators.

The MathStation Programming Language is based on a WYSIWYG editor. MPL includes loops, conditionals, functions, procedures, macros, and other programming constructs for building algorithms and complex models.

The software permits integration of typeset mathematics with text and graphics, and produces PostScript output. The interface is based on the X-

Windows graphical environment from MIT. MathStation requires the X.11 version of X-Windows.

According to the company, the software can be used to build and test numerical applications before Fortran code is generated. Users can integrate their existing Fortran routines and libraries for use in their MathStation documents. The C-based software requires a Fortran 77 compiler.

Remote compute servers, available separately, permit users to send complex or CPU-intensive computations to Cray or VAX parallel computers for processing.

Prices for MathStation start at \$9,500 for a Sun-3-based, single-user license on a network. Prices vary according to configuration and volume.

Reader Service 52

Zuken tackles CAE with new systems

Zuken America, a subsidiary of Zuken Inc. of Japan, offers a series of products targeted at computer-aided engineering design. They include the CR 3000, the Flex Router, the Analog Workstation, and the Logic Workstation.

The CR 3000 is a gridless electrical CAE/CAD/CAM system that reportedly features comprehensive digital, hybrid, and analog design capabilities with a unified database structure. The system runs on the Hewlett Packard Model 350, a 32-bit Unix-based workstation. The software comes unbundled or as part of a turnkey system. Prices range from \$70,000 to \$140,000. A typical configuration costs about \$115,000.

The Flex Router is a gridless software program for automatically routing ultra-high-density circuits. It runs on Zuken's CR 3000 system. It also functions as a stand-alone workstation

engine using industry standard workstations. A stand-alone system costs around \$25,000 depending on the hardware.

The Analog Workstation is a CAE system for designing analog circuits, automatically generating circuit networks, and graphically analyzing circuits. Software packages available include schematic capture, design rule checking, and HSPICE circuit simulation. A fully configured CAE system costs about \$80,000.

The Logic Workstation features a logic simulator for designing and evaluating logic circuits. The software costs \$10,000 and runs on a Unix platform.

CR 3000: **Reader Service 53**
Router: **Reader Service 54**
Analog: **Reader Service 55**
Logic: **Reader Service 56**

POSE comes with money-back guarantee

Computer Systems Advisors offers POSE (Picture Oriented Software Engineering) software with a money-back guarantee in an attempt to capture part of the CASE market.

According to the company, POSE features nine PC-based modules for the planning, design, and analysis phases of application-software development using either data-driven or process-driven approaches. Each module reportedly employs a Macintosh-like user interface and facilities for data validation, integration, and normalization.

Currently available modules are Data Model Diagrammer, Data Model Normalizer, Logical Database Designer, and Data Base Aid (which make up the data-driven toolkit); and DeComposition Diagrammer, Data Flow Diagrammer, Structure Chart Diagrammer, and Action Chart Diagrammer (which make up the process-driven toolkit). The screen and report prototyping module is scheduled for fall delivery.

POSE runs on IBM PC, XT, AT, PS/2, and compatible computers. It supports SQL/DS, DB2, and Adabase.

Individual modules have introductory prices of \$295 each. The toolkits cost \$885 each.

Reader Service 57

Software simulates a neural circuit array

Abbot, Foster & Houserman Co. offers The Brain Simulator: Tutorial Software for Neural Circuit Design. The software simulates a neural circuit array on MS-DOS computers.

According to the company, by emulating the electronic function of the biological neuron, the program teaches users about the function of the neuron and how it is incorporated into the circuitry of the brain. The program includes examples of the types of circuitry that can be created with the neural model.

The Brain Simulator implements an array of 1,200 neurons using a digital model of the human brain neuron developed by the Neural Network Laboratory. Users can modify the sample networks or define new ones.

The Brain Simulator costs \$99.

Reader Service 58

Apollo builds new personal workstations around MC68030 chip

Apollo Computer has announced the Series 4500 Personal Super Workstation and the Series 3500 Personal Workstation, both based on the Motorola 68030 chip. The company claims that the new workstations also incorporate a series of innovative system design features, including a 64-Kbyte physical cache, interleaved memory, and no-wait-state operation.

The Series 4500 incorporates a 33-MHz 68030 chip and 68882 floating-point processor. It reportedly achieves more than 7 million instructions per second. Prices start at less than \$19,000.

The Series 3500 features 25-MHz 68030 and 68882 floating-point processors, and 4-32 Mbytes of main memory. It reportedly achieves 4 MIPS. Prices for the monochrome workstation start at less than \$8,000 and for the color workstation, at less than \$10,000.

The company offers upgrades from the Series 3000 and Series 4000 to the Series 3500 and Series 4500, as well as an upgrade from the Series 3500 to the Series 4500.

Series 3500: **Reader Service 59**
Series 4500: **Reader Service 60**

Alliant launches Tempest system

Alliant Computer Systems has announced the FX/80T Tempest system. (Tempest systems limit the emission of intelligence-bearing electromagnetic signals by information-processing systems. The National Security Agency created the term.)

The FX/80T meets National Communications Security Information Memorandum (Nacsim) 5100A specifications. The system is field-upgradable from two to eight vector processors. According to the company, the fully configured FX/80T provides 188.8 peak Mflops and 118 Whetstone MIPS of computational throughput.

Alliant plans to ship the FX/80T in

the fourth quarter of 1988. The system was jointly developed with the Tempest Division of Atlantic Research, which is performing the certification testing.

The FX/80T base configuration costs \$449,000. It includes two vector processors, two VME-based interactive functions, 32 Mbytes of memory expandable to 256 Mbytes, a 16-line multiplexer, 1.1 Gbytes of fixed or removable disk storage, and a tridensity magnetic tape drive. The system also includes Ethernet support, Tempest console terminal and printer, a 16-user Concentrix operating system license, and FX/Fortran.

Reader Service 61

DECtp builds transaction processing applications

Digital Equipment has introduced DECtp, a systems environment for the building of transaction processing applications. The system includes databases, storage systems, data interoperability products, transaction processing monitors, and support programs.

According to the company, DECtp includes a transaction processing monitor, DECintact; higher-performance versions of the VAX Rdb/VMS relational database management system and its supporting software; DEClink software for access to IBM-based databases; and the SA600 Storage Array with a 9.7-Gbyte capacity.

A typical DECtp system based on a MicroVAX 3500 with DECintact costs

about \$380,000. A relational database system using a VAX 6220 and VAX ACMS/Rdb costs about \$967,000. A VAX 8842-based system with DECintact and SA600 costs about \$3,883,000.

Used with DEC's HSC70 storage servers, four SA600 units can be tied together. The system can be added to systems already using DEC's other storage products.

The SA600 comes in two configurations. The fully configured 9.7-Gbyte system costs \$225,000. The partially configured 4.8-Gbyte system costs \$125,000.

Reader Service 62

Expert system shell claims connectivity

Programs in Motion offers an expert system shell called 1st-Class Fusion, which the company claims offers the capabilities of 1st-Class plus advanced internal functionality and enhanced external connectivity with other hardware and software systems.

According to the company, 1st-Class Fusion's code generators produce program source code in C or Pascal, or production rules. Graphics capture and display utilities permit users to capture screen shots from other programs. A math calculator permits incorporation of math operations into knowledge bases, while a dBase III interface allows users to access and update DBF files directly from a running advisor.

A tracer facility assists testing and debugging, while a road map facility shows all the chains to a knowledge base for planning and building large, chained expert systems.

1st-Class Fusion comes with a runtime program. No further royalties are required for its distribution.

1st-Class Fusion costs \$1,295.

Reader Service 63

OEM systems based on iRMX II OS

Intel has announced a family of original-equipment manufacturer computer systems based on the iRMX II real-time operating system. The single-processor System 120 models are reportedly designed for real-time applications that can use a 386-based, AT-bus hardware platform.

According to the company, applications developed using the System 120 are upwardly compatible with Intel iRMX II-based Multibus I and Multibus II systems.

The System 120 comes in either a development or target system configuration. The platform version comes in two models, with and without disk drives. The development version is available on the hard disk drive models only. It includes either a System 120 or Multibus system developer's toolkit.

The platform configuration prices begin at \$3,800 in quantities of 50. A development system configuration costs \$9,250 and up for single units.

Reader Service 64

IC Announcements

Company, Model, Function	Comments	R.S. No.
Advanced Micro Devices Am79101 Modem	An autodial, single-chip, frequency-shift-keying modem that operates at 300 bps full-duplex and 1,200 bps half-duplex. Provides modulation, demodulation, filtering, A/D, and D/A functions on-chip. Comes in 28-pin PLCCs and plastic and ceramic DIPs. Cost: \$17.60 (100s).	120
Analog Devices 2S82 R/D converter	A resolver-to-digital converter, pin-programmable for 10-, 12-, 14-, and 16-bit resolution. Bandwidth, max tracking rate, and velocity scaling set by user. Connects directly to a 16-bit or 8-bit data bus. Four grades, based on accuracy. Comes in a 44-terminal PLCC. Cost: \$47.19 for HP grade to \$77.84 for JP grade (1,000s).	121
Analog Devices AD845 Op amp	An FET-input op amp with a typical settling time of 350 ns to 0.01% (for a 10V step) into a 500 Ω /100 pf load. Total harmonic distortion below 0.0001%. Comes in plastic 8-pin miniDIP and hermetic ceramic DIP over three temperature ranges. Cost: starts at \$9.50 for K grade (100s).	122
Inmos IMS A110 DSP	A DSP chip that can execute 400 MOPS on video data at speeds up to 20 MHz. Features a parallel-processing architecture. Digitized data flow through a pipeline of 21 multiplier/accumulators. Comes in a 100-pin ceramic PGA. Cost: about \$500 for samples.	123
Integrated Device Technology IDT10490 SRAM	A 64K $\times$ 1 static RAM compatible with ECL-10K logic levels. Consumes 120 mA to deliver 10 ns. Features a min 8-ns write pulse width, open emitter output, separate data input and output, and static operation. Comes in a 22-pin ceramic DIP. Cost: \$88.50 (100s).	124
Integrated Device Technology IDT72021/31/41 FIFOs	FIFOs with multiple flags and an output enable. All three feature almost-empty and almost-full flags. 72021 (1K $\times$ 9) has a 25-ns access time. 72031 (2K $\times$ 9) and 72041 (4K $\times$ 9) have a 35-ns access time. They come in a 32-pin plastic DIP, ceramic DIP, PLCC, and CLCC. Cost: (100s) \$49 for plastic DIP 72021; \$65 for 72031; \$98 for 72041.	125
Motorola 68882 Math coprocessor	A 33-MHz version of the 32-bit, floating-point math coprocessor. Conforms to the IEEE Standard for Binary Floating-Point Arithmetic and is software and pin compatible with the 68881 floating-point math coprocessor. Exceeds two million Whetstones in processing speed. Cost: \$708.	126
Sierra Semiconductor SC11013 Controller	A single-chip, ASIC controller that supports Microcom Networking Protocol, service classes 2, 3, and 4. Used with the SC11006, a 2,400-bps modem analog peripheral. Also requires a type 8530 USART, a type 6264 8K $\times$ 8 SRAM, and some discrete logic. Comes in a 68-pin PLCC. Cost: \$70.70 for kit with SC11006 modem (100s).	127
Signetics PLS173D PLA	A programmable logic array with a worst-case propagation delay of 12 ns. Based on the PLS173 family architecture. Comes in a plastic or ceramic DIP and PLCC package. Cost: (in production quantities) \$10.50 for plastic DIP; \$11.50 for PLCC.	128
Signetics PLHS153, PLUS153B/153D PLAs	Worst-case propagation delays of 20 ns (PLHS153), 15 ns (153B), and 12 ns (153D). Additions to the PLS153 family of 20-pin combinatorial devices. Come in plastic and ceramic DIPs and PLCCs. Cost: \$3.75 for PLHS153 plastic DIP, \$4.25 for PLCC; \$5 for PLUS153B plastic DIP, \$5.50 for PLCC; \$6.50 for PLUS153D plastic DIP, \$7 for PLCC.	129
Silicart GSC Graphics chip	A graphic system chip featuring a RISC graphic processor, a drawing accelerator with a max 10 million pixels/s drawing speed, and a video controller. Can execute formats up to 1,024 $\times$ 768 pixels, including IBM's VGA. Cost: about \$100 (1,000s).	130
Thomas-Conrad TC9021 ROM	A remote reset ROM compatible with Novell NetWare 2.0A and 2.1x. Permits implementation of diskless workstations on the Arcnet LAN. Supports multiple boot image files and includes the configured Novell workstation software. Compatible with DOS versions 3.1, 3.2, and 3.3. Requires 256 Kbytes of memory. Cost: \$40.	131
Zilog Z84C01 Microprocessor	An 8-bit microprocessor that combines a 10-MHz Z80 CPU and a clock generator/controller. Includes all 158 Z80 instructions. Accommodates four kinds of operating modes in connection with clock generation. The clock oscillator circuit accepts either crystal or TTL-compatible clock source inputs. Cost: \$4.50 (100s).	132

Students, Membership Is Part Of Being A Professional...Join The Computer Society Now!

As a student member of the Computer Society, you will have a wealth of resources for professional success, both in school and beyond. The student cost to join is much less than you'd think—and the benefits, much greater. Among the many automatic benefits offered to IEEE/Computer Society students are,

- ☐ Computer magazine monthly
- ☐ IEEE Spectrum, The Institute, and Potentials
- ☐ Local chapter activities
- ☐ Discounts on optional periodicals, Computer Society Press books, and conferences.

1

PERSONAL

Name _____

Street address _____

City _____

State/Country _____

Mail code _____

☐ Male ☐ Female Date of birth _____ / _____ / _____
month year day

NOTE: Students MUST be IEEE members to qualify for membership in the Computer Society. If you are not already an IEEE member, you may sign up for both memberships at once. To qualify for student membership, we must receive your application prior to graduation.

2

EDUCATION & ENDORSEMENT

School name _____

City _____

State/Country _____

Mail code _____

Degree expected _____ Expected grad. date _____

Undergraduate: ☐ 2-3 year ☐ 4-5 year/Graduate: ☐ Masters ☐ PhD
 Degree sought: ☐ EE ☐ CS ☐ Comp. Eng. ☐ Physics ☐ EE Tech. ☐ Other _____

(If there is no chapter advisor at your school, please obtain the endorsing signature of a faculty member who is also an IEEE member, and include membership number.)

Name of IEEE advisor/counselor _____

School code _____

I endorse this application for student membership in the IEEE and the Computer Society, and certify the degree information as given to be correct and complete.

Signature of IEEE advisor/counselor _____

Date _____

Member no. _____

3

FEES

Membership dues are annualized to, and expire on, December 31. Choose full or half-year rate schedules depending on date of receipt by the Computer Society as indicated below. Please check the membership category that describes your status at the time of this application.

	Full Year 9/1-2/28	Half Year 3/1-8/31
I am an IEEE student member* who wishes to join the Computer Society	☐ \$ 8.00	☐ \$ 4.00
I am a student who wants to join BOTH the IEEE and the Computer Society	☐ 28.00	☐ 14.00

*IEEE no. _____

☐ Student members in the U.S. and Canada automatically receive *Potentials* magazine. Students residing outside these areas may elect not to receive *Potentials* by checking here and subtracting \$6 from membership fees.

TOTAL ENCLOSED \$ _____

Checks are accepted in Belgian, British, German, Japanese, Swiss, or U.S. currencies. U.S. checks must be drawn on a U.S. bank.

☐ Check/Money Order ☐ Visa
☐ American Express ☐ Mastercard ☐ Eurocard

Card number _____

Expiration date _____

Return with your remittance to: The Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720 USA. Residents of Europe mail to: The Computer Society, 13, Avenue de l'Aquilon B-1200 Brussels, BELGIUM. Asian region residents mail to: The Computer Society, Ooshima Building, 2-19-1 Minami-Aoyama, Minato-ku, Tokyo 107, JAPAN.

AGREEMENT

I hereby apply for IEEE Computer Society student membership and, if elected, will abide by the IEEE constitution and bylaws, and its statements of policies and procedures.

Student's signature _____

Date _____



THE COMPUTER SOCIETY

THE INSTITUTE OF ELECTRICAL
AND ELECTRONICS ENGINEERS, INC.

Nonstudents interested in membership:
see regular application elsewhere in this issue.

Microsystem Announcements

Company, Model, Function	Comments	R.S. No.
Applied Microsystems ES 1800 Emulator	A zero-wait-state, 16.7-MHz emulator for the Motorola 68000. Operates with Sun, Apollo, VAX, IBM PC, and other computers. Comes with the Advanced Event Monitor System. A probe module supports both the DIP and PLCC 68000 packages. Cost: starts at \$11,000; \$4,095 for probe module.	135
Arcom Control Systems STELA Logic analyzer	A logic analyzer for STEbus activity. Consists of a standard Eurocard with a front-panel board housing switches and displays. Can be set to trigger on any kind of STEbus access or particular STEbus cycles. A trigger-output signal activates an analysis instrument if needed. Cost: £355.	136
Ariel DSP-300 Coprocesor	A signal-processing coprocessor for Hewlett Packard 9000 Series 200/300 computers. Combines Motorola's DSP56001 CPU with up to 128 Kwords of zero-wait-state memory. Supports DMA and interrupts. Comes with Basic and Pascal interfaces with source code. Occupies a DIO card slot. Cost: \$2,400.	137
Caplin Cybernetics QT1, QT2 Transputer boards	Two boards to equip DEC MicroVAX II or 3000 series computers with transputer networks, coupled to the Q-bus. The boards come equipped with IMS T414-15, T414-20, T800-17, or T800-20 transputers and will accept the IMS T800-30 devices when available. No prices given.	138
DY-4 Systems Internetworking Protocol Suite TCP/IP firmware	An on-board TCP/IP firmware package for the VMEbus. Consists of TCP/IP, a Motorola S-Record server, Telenet server, monitor, and diagnostics. Available on the DY-4 VMEbus DVME-750 IEEE 802.3 LAN Controller. No price given.	139
Gemini Technology Omega 800 Plus Graphics board	An EGA graphics board for IBM PC, XT, and AT computers. Uses the VC-001 chip, a video controller. Comes with utilities; drivers; standard EGA, CGA, MDA, and Hercules text modes; and 24 combinations from 80-132 columns by 21-74 rows. Cost: \$299.	140
Hewlett Packard HP ROM disc accessory ROM card	A ROM card for HP Vectra PCs. Can be set up to load applications automatically in ROM. Comes with MS-DOS 3.2 and sockets for up to 768 Kbytes of memory. Available with a programming kit. Features a protection device to prevent copying of software installed on the card. Cost: \$235 for card; \$260 for kit.	141
MetraByte MBC-625 Data acquisition board	A data acquisition board for Macintosh II computers. Consists of a multifunction NuBus-compatible base board that can accept up to three daughter boards. An expansion system, the MBC-SE-625, provides the interface and power supply to connect the board to Macintosh SE computers. Cost: \$1,290 for board; \$1,890 for expansion system.	142
National Semiconductor VME532 boards Board-level computers	New models, in 20-, 25-, and 30-MHz versions. Combine the NS32532 CPU with Weitek's WTL3164 floating-point data-path component. Volume production in fourth quarter. Cost: \$5,740 (100s) for the 20-MHz version with 64 Kbytes of cache memory and 4 Mbytes of main memory.	143
Newer Technology LIMfile Memory board	A 16-bit memory expansion board with LIM 4.0 support and a nonvolatile RAM disk. Features multiple hardware register support. Comes in volatile and nonvolatile versions. Compatible with AT-bus systems at 6-24 MHz. Memory up to 8 Mbytes per board. Daughter boards optional. No prices given.	144
Quam QC 1048 Board-level computer	An AT-compatible board based on the Intel 80286. Operates at 10 or 12 MHz. On-board DRAM expandable up to 4 Mbytes with single or zero-wait-state modes. Includes two serial channels and a printer port. Comes with Quam's ROM BIOS. Cost: \$589 (10 MHz) and \$659 (12 MHz) for zero RAM.	145
Silicon Composers SC/Fox Coprocesor	A general-purpose, parallel coprocessor based on the Harris Forth-based RTX 2000 real-time processor. Plugs into IBM PC, XT, AT, 386, or compatible computers. Has a 16-MHz board clock upgradable to 20 MHz, 64 Kbytes of 45-ns SRAM upgradable to 1 Mbyte, and two 50-pin application connectors. Comes with a software development system. Cost: \$1,995 (64 Kbytes, 8 MHz) to \$3,495 (256 Kbytes, 10 MHz).	146
Xycom XVME-602 Processor module	A VMEbus-compatible processor module based on a 16-MHz 68020 CPU. Comes with 256 Kbytes of SRAM and sockets for up to 256 Kbytes of EPROM. Capable of 32-bit accesses on-board. Can access the VMEbus in the A24 or A16 address space. Has two RS-232 serial ports, a 16-bit timer, and an expansion connector. Cost: \$1,650.	147



CALL FOR PAPERS

26th ACM/IEEE DESIGN AUTOMATION CONFERENCE®

DAC 89 is the premier conference devoted solely to the field of Design Automation. All aspects of the use of computers as aids to the design process, from conceptual design through manufacturing, are included. Four session types are included: regular paper sessions, short paper sessions, panels, and tutorials

REQUIREMENTS FOR SUBMISSION OF PAPERS

Authors should submit their papers to the program chairman no later than November 7th, 1988. Each submission should include one cover page and eight stapled copies of the complete manuscript. The one cover page should include:

- Name(s), affiliation(s), complete address(es), identification of principal author, and telephone number.
- The following signed statement: "All appropriate organizational approvals for the publication of this paper have been obtained. If accepted, the author(s) will prepare the final manuscript in time for inclusion in the conference proceedings and will present the paper at the conference."

The eight copies of the complete manuscript should each include:

- Title and abstract page: title of paper, 60-word abstract indicating significance of contribution, and an ordered list of number(s) from the category(s) at the right that most closely match the content of the paper. To permit blind review, do not include name(s) of author(s) on this page.
- The complete text of the paper in English, including all illustrations and references, not exceeding 5000 words. The length of the paper is considered in the review process.

Notice of acceptance will be mailed to the principal authors by February 15th, 1989. Authors of accepted papers must sign a copyright release form.

TOPICS OF INTEREST

Authors are invited to submit original technical papers describing recent and novel research or engineering developments in all areas of design automation. Topics include, but are not limited to:

- 1.1 Electrical Simulation
- 1.2 Discrete Simulation
- 1.3 Timing Verification
- 2.1 Test Validation and Testability Analysis
- 2.2 Test Pattern Generation and Design-for-Test
- 2.3 Formal Methods for Design Verification
- 3.1 Floorplanning and Placement
- 3.2 Global and Detailed Routing
- 3.3 Physical Module Generation
- 3.4 Symbolic Layout and Compaction
- 3.5 Complete Layout Systems
- 3.6 Layout Verification (DRC, ERC)
- 4.1 Logic Synthesis and Optimization
- 4.2 Register-Level Synthesis and Optimization
- 4.3 Behavioral Synthesis and System-Level Design Aids
- 5.1 Behavioral and Hardware Description Languages
- 5.2 Design Systems and Databases
- 6.1 DA for IC Fabrication
- 6.2 Computer Aids for Manufacturing
- 7.1 DA for Analog Circuits
- 7.2 High Speed Systems and Microwave DA
- 7.3 DA for Electronic Packaging
- 8.1 Human Factors in DA
- 8.2 Software Engineering in DA

PANELS, TUTORIALS, AND SPECIAL TOPIC SESSIONS

Proposals for topics of Panels, Tutorial Sessions, and Full-Day Tutorials should be submitted to the program chairman no later than November 7th. Each proposal should not exceed two pages in length and should include a description of the topic, structure of the session or tutorial, and a list of participants. Contact should have been made with all participants in advance.

Special Topic Sessions may be either independent papers with a common theme or a set of closely related papers describing an overall system. In both cases, independent reviews of each paper and evaluation of the session as a whole will be used to select sessions. Proposals for Special Topic Sessions should be submitted along with the papers to be included in the session and should describe the theme of the session. These proposals and papers must be submitted by November 7th to allow for complete review of the papers.

PROGRAM CHAIR

MP Associates, Inc.
ATT: A. Richard Newton
Program Chairman 26th DAC
7490 Clubhouse Road, Suite 102
Boulder, Colorado 80301
For information call: (303) 530-4333



LAS VEGAS CONVENTION CENTER JUNE 25-29, 1989



CAREER OPPORTUNITIES

RATES: \$12.00 per line, \$120 minimum charge (up to ten lines). Average five typeset words per line, eight lines per column inch. Add \$10 for box number. Send copy at least one month prior to publication to: **Heidi Rex or Marian Tibayan, Classified Advertising, COMPUTER Magazine, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; (714) 821-8380.**

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *COMPUTER* may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...", "...1-4 years maximum experience...", "...up to 5 years experience...", or "...10 years maximum experience." *COMPUTER* reserves the right to append to any advertisement, without specific notice to the advertiser, "Experience ranges are suggested minimum requirements, not maximums." *COMPUTER* assumes that, since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only.

CENTRE DE RECHERCHE INFORMATIQUE DE MONTREAL - CRIM

Computer Communications Researcher

Duties: Within the context of Actions Structurantes, CRIM seeks applicants for research position to work with its project team in Computer Communications. Work in Computer Communications is focussed on LAN's, fiber technology, voice-data integration, internetworking, performance evaluation, communication "software", formal specifications, protocol development environment, applications to office automation and distributed processing, voice and graphic interfaces video-text and data bases.

Opportunity to teach in one of the universities and/or to participate in one of the industrial research projects.

Requirement: Ph.D. in computer science with research expertise relevant to the interest of the team.

Agreement: One (1) year renewable. This position is effective on October 2, 1988. Applicants should send a resume, list of publications and two (2) references to:

Dr. Jacqueline Bourdeau
Assistant Director of Scientific Research
Reference number: DS880711-A
Centre de recherche informatique
de Montreal
1550, de Maisonneuve West Blvd.
Suite 1000
Montreal, Quebec
H3G 1N2

PENN STATE Computer Engineering

Applications are invited for faculty positions at all levels. Candidates from all areas of computer engineering (hardware and software) will be considered. The Computer Engineering Program at The Pennsylvania State University is within the Department of Electrical Engineering which has over 50 faculty members, and approximately 1300 undergraduate majors, 200 graduate students. Candidates should have a Ph.D. in Electrical/Computer Engineering or related areas. There are 13 faculty members within the Computer Engineering Program. Excellent instruction and research computing facilities are available within the Department, College and at the University Computation Center.

Send letter of application, resume, or inquiries, together with three references to: **COMPUTER ENGINEERING PROGRAM, DEPARTMENT OF ELECTRICAL ENGINEERING, 129 ELECTRICAL ENGINEERING EAST Box-C, THE PENNSYLVANIA STATE UNIVERSITY, UNIVERSITY PARK, PA 16802.**

Deadline for applications is November 30, 1988, or until suitable qualified candidates are selected.

An Affirmative Action/Equal Opportunity Employer Women and Minorities Encouraged to Apply.

NAVAL POSTGRADUATE SCHOOL Computer Science Chairperson

The Naval Postgraduate School invites applications and nominations for the position of Chairperson of the Computer Science Department. We are seeking an individual who can provide leadership in both teaching and research to a strong and growing program. This person must have a Ph.D. in Computer Science or a closely related area and a distinguished record of research and publication as well as demonstrated administrative capability.

The Department offers M.S. and Ph.D. degrees in Computer Science. Currently, 110 students are enrolled in the M.S. program and five in the Ph.D. program. No undergraduate degrees are offered. Students are military officers or civilian employees of the Department of Defense and are fully supported by their sponsoring organization during their graduate studies.

The faculty currently consists of thirteen full-time civilian faculty, five military faculty, and several adjunct faculty. Additional positions are open in all areas. Departmental facilities (supported by eight full-time computer professionals) consist of six instructional and research laboratories including the Database Systems Laboratory, the Graphics and Video Laboratory, the Software Engineering Laboratory, the Microcomputer Systems Laboratory, the Artificial Intelligence Laboratory, and the Computer Science Academic Laboratory.

In these laboratories are state-of-the-art computing equipment including a VAX 780, a VAX 785, LISP machines, high-performance graphics workstations, bit-mapped graphics workstations, and multi-microprocessor systems.

Located on Monterey Bay, the areas of Monterey, Pebble Beach, and Carmel provide a pleasant Northern California coastal climate and easy access to Silicon Valley companies and universities.

Please send a detailed resume and letters of reference to:

Chair Search Committee
Computer Science Department (Code 52)
Naval Postgraduate School
Monterey, CA 93943
Tel. No. (408) 646-2449
AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER.

UNIVERSITY OF KENTUCKY Chaired Professorship in Computer Engineering

The Department of Electrical Engineering at the University of Kentucky is seeking a nationally recognized scholar to fill the Robinson Chair in Computer Engineering. Qualifications include a Ph.D. in Electrical Engineering and an ability to establish, develop extramural support for and maintain a viable and nationally recognized research program in Computer Engineering. Also required is an ability to lead development of and teach, where appropriate, supporting graduate level courses in Computer Engineering. This position offers outstanding professional opportunities as well as a highly competitive salary. A major priority of the University of Kentucky is the establishment of Computer Engineering as a nationally recognized Center of Excellence with the Robinson Chair providing the scholarly leadership required to develop and maintain such a Center. Additional existing support of the position includes a university administration supportive of computing activities, six existing faculty positions in Computer Engineering, excellent laboratory and office space, supporting computing facilities including an on-campus supercomputer, mainframes, departmental minicomputers, microcomputers, and office workstations, all connected through a campus-wide data communications network. Supporting units of interest include a Computer Science Department, a recently established Center for Computational Sciences and the University Computing Center. Nominations or applications should be sent to Dr. J. Robert Heath, Chairman, Robinson Chair Search Committee, Department of Electrical Engineering, University of Kentucky, Lexington, KY 40506-0046. Additional information may also be obtained by calling (606) 257-3124. The University of Kentucky is an equal opportunity/affirmative action employer.

UNIVERSITY OF SOUTH FLORIDA
Computer Science and
Engineering Department

The Department of Computer Science and Engineering invites applications for faculty positions at the Assistant Professor, Associate Professor, and Full Professor levels. The Department is approximately 8 years old, offers Bachelors, Masters and Ph.D. degrees in both Computer Science and Computer Engineering. The Department is housed in a \$10,000,000 new building and the size of the faculty is scheduled for rapid expansion over the next years. Current areas of research emphasis in the department include Computer Architecture/VLSI Design & Test, Computer Vision/Graphics; Image Processing, Artificial Intelligence, and Software Engineering. The departmental faculty participate in microelectronics Design and Test activities supported by the college of Engineering and the State of Florida.

Applicants for positions are especially invited in the area of Computer Architecture, Distributed Computing/Networking, Computer Vision/Graphics/Image Processing, Artificial Intelligence and Expert Systems, Automated VLSI Design/Hardware-Software Interaction, and Software Engineering. Exceptional applicants in other areas are welcome.

The University of South Florida is the second largest University in Florida, with a current enrollment of over 25,000. It is located on a 1,700 acre campus at the northeast edge of the city of Tampa. The Tampa metropolitan area is one of the largest and fastest growing in Florida, offering a wide variety of cultural, entertainment, and sports activities. The cost of living is moderate and the quality of life is excellent. The area surrounding the University and its Medical School is experiencing dramatic growth in industry and medical facilities. A variety of firms are located near the campus, including Honeywell, Sperry, IBM, GTE Data Systems, E-Systems, and AT&T. A PBS public-TV channel is located on the campus; other television links tie the University to industry and remote campuses. Applicants should send a resume and the names of three references to: Faculty Search Committee, Computer Science and Engineering, University of South Florida, Tampa, FL 33620. USF is an equal opportunity and affirmative action employer.

ROCHESTER INSTITUTE
OF TECHNOLOGY
School of Computer Science

Visiting tenure track faculty position available in September or December for individual who has experience in the instruction of Computer Science or Software Engineering (preferred). Ph.D. or Master's (with professional experience) degree in appropriate discipline required. Duties will include teaching at the undergraduate and graduate (M.S.) level as well as applied research (Software Engineering preferred).

Send resume to Evelyn Rozanski, Director, School of Computer Science, 1 Lomb Memorial Drive, Rochester, New York, 14623. RIT is an equal opportunity affirmative action employer.

TRINITY COLLEGE

The Department of Engineering and Computer Science invites applications from outstanding candidates for a tenure-track position at the Assistant Professor-level commencing September, 1989, in the areas of FLUID MECHANICS/HEAT TRANSFER or ROBOTICS/CONTROLS. Experimental background highly desirable. The position involves graduate and undergraduate instruction and research, and a doctoral degree is a prerequisite. We are interested in receiving applications from qualified women and minorities. Trinity College is an equal opportunity/affirmative action employer. Please send resume to Professor Joseph D. Bronzino, Chairman, ECS Department, Trinity College, Hartford, CT 06106. Applications will be accepted until February 15, 1989.

NAVAL POSTGRADUATE SCHOOL
Position Announcement in
Computer Science

The Department of Computer Science has immediate openings for faculty positions at all levels. Our primary interests are in the areas of operating systems, distributed systems, programming languages, and algorithms. Our secondary interests are in the areas of processing of visual data, real-time systems, and software engineering. An applicant should have a PhD in Computer Science or a related field and have a strong interest in both graduate teaching and research. Senior applicants must have distinguished research records. Appointments can begin at any time during the year.

The Department offers MS and PhD degrees in Computer Science. Departmental facilities (supported by 9 full-time computer professionals) consist of six instructional and research laboratories including the Database Systems Laboratory, the Graphics and Video Laboratory, the Software Engineering Laboratory, the Microcomputer Systems Laboratory, the Artificial Intelligence Laboratory and the Computer Science Academic Laboratory. In these laboratories are the latest, state-of-the-art computing equipment including LISP machines, high-performance graphics workstations, bit-mapped graphics workstations, multi-microprocessor systems, etc. During the academic year, the faculty normally teach for two quarters and conduct full-time research supported by major research programs in both the Department of Defense and the non-DOD organizations during the other two quarters. New faculty receive initial support from an on-campus research foundation.

Located on Monterey Bay, the areas of Monterey, Pebble Beach, and Carmel provide a pleasant Northern California coastal climate and easy access to Silicon Valley companies and universities.

Please send a detailed resume and three letters of reference to:

Search Committee
Computer Science Department (Code 52)
Naval Postgraduate School
Monterey, CA 93943
Tel. No. (408) 646-2449

AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER.

UNIVERSITY OF CALIFORNIA, IRVINE
Department of Electrical Engineering
Computer Engineering Area

The Department of Electrical Engineering at the University of California, Irvine, is continuing to expand its Computer Engineering capability. We are seeking an exceptional senior faculty member to be hired at the full-professor level, to complement the strengths we have in advanced computer architectures, systems software, and advanced applications of computer engineering, including computer communications, artificial intelligence, and others. This senior position would have responsibilities for research and teaching at the undergraduate and graduate level, as well as assisting in the guidance and development of the computer engineering thrust area. Candidates should have an established research and funding record.

The Department has research activities in design automation, VLSI design, distributed computing, fault-tolerant computing, and real-time computing, as well as digital signal processing, image understanding, pattern recognition, robotics, digital communications, and communication networks. We currently have twenty-two faculty members, two of whom are members of the National Academy of Engineering and eight of whom are Fellows of the IEEE.

UCI is located in one of the fastest growing areas in the country. Situated in Orange County, the heart of a national center of high-technology enterprise, both the campus and the surrounding community are experiencing substantial growth with exciting opportunities. Numerous research and technology organizations in this area provide occasions for collaborative research efforts.

Irvine is three miles from the Pacific Ocean, and approximately half way between Los Angeles and San Diego. Special housing assistance is available from the university, including newly built for-sale housing within short walking distance from the school. Salaries and benefits are highly competitive.

Please send your resume and the names of four references to: Jose B. Cruz, Jr., Chair, Department of Electrical Engineering, University of California, Irvine, CA 92717.

Apply before December 1, 1988 to be sure of full consideration. We are an affirmative action/equal opportunity employer, and women and minorities are strongly urged to apply.

D.E. SHAW & CO.

Futuristic new investment bank seeks several of the world's greatest system designers to build the most advanced computer environment on Wall Street. Computer scientists will form the professional core of our firm, and not merely its support staff, and will participate in its profits. Relevant expertise includes Unix, C, Sun, Ethernet, Sybase, distributed real-time applications, and data communications. Exceptional academic and professional credentials or other evidence of extraordinary competence is required. We are prepared to compensate unusually gifted individuals at a level exceeding that of the market. Resume to D.E. Shaw & Co., 13 E. 16th St., 6th Floor, New York, NY 10003.

SOFTWARE R&D

International Software Systems, Inc., is a technology and research company located in Northwest Austin. Our efforts are focused on creating software environments and tools that dramatically improve productivity in developing complex software applications. Part of our efforts are funded by government contracts which promise high visibility of our results.

We have needs for individuals with research and development skills in the following areas:

- Software Development Methodologies
- Object Oriented Data base
- Graphics-Oriented User Interfaces
- Expert Systems • Formal Languages
- Automatic Code Generation

Work will be done using UNIX on Sun Workstation. Candidates must have a strong technical background, extensive research or design and development experience in one or more of the above mentioned areas, and should be creative thinkers. Advanced degree in Computer Science, Physics, or Engineering is highly desirable.

Please send your resume to:
International Software Systems, Inc.
Personnel Dept. DWH
9420 Research Blvd.
Austin, TX 78759

PURDUE UNIVERSITY Computer Engineering Faculty Positions

The School of Electrical Engineering at Purdue University seeks outstanding candidates in all areas of Computer Engineering for research and teaching. Openings are for tenure-track faculty at all levels. Active research areas include artificial intelligence and expert systems; computer communication networks; computer vision; design automation tools; digital signal processor system design; distributed algorithms and databases; fault-tolerant and testable computing; microprocessor design; neural networks; parallel processing (architecture, algorithms, operating systems, compiling, languages, interconnection networks, and performance); robot vision, sensors and control; software engineering; speech processing; and VLSI architecture.

The School has 65 full-time faculty (26 in Computer Engineering), and over \$8.5M in funded research projects. In addition to the PhD, MSEE, and BSEE degrees, the School offers a BSCEE (Bachelor of Science in Computer and Electrical Engineering) which is accredited in both Computer Engineering and Electrical Engineering. Computing facilities available to the faculty include the Engineering Computer Network (including 17 VAX 11/780's running UNIX BSD 4.3, 1 Gould PN 9080, 4 Gould NP 1's, a CCI PN 6/32, and 225 Sun workstations), the Computing Center's Cyber-205 and ETA-10 supercomputers, a Symbolics LISP machine, and IBM 3090/180E, extensive graphics facilities, and numerous PC's. Parallel computing facilities include a 128-node Ncube hypercube, a 48 x 48 processor NCR-GAPP processor array, a 10 processor Transputer array, the 30-processor PASM Parallel Processor prototype that

was developed and built at Purdue, and the Computing Center's Sequent Balance 21000. Custom VLSI chip facilities include an IBM Master Image System. Equipment in the Robot Vision Lab and Computer Vision and Image Processing Lab includes a Puma 762 robot, a Cincinnati Milacron T3-726 robot, a K2A Cybermation mobile robot, and Gould/DeAnza, Comtal, Grinnell, and Imaging Technologies image processing systems. Support facilities provided by the School include a technical typing pool and a graphics illustrator.

Applicants must possess a doctorate degree. Send a resume, including a statement of teaching and research interests and a list of at least three references, to: Head, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907. Purdue University is an Equal Opportunity/Affirmative Action employer.

COLORADO STATE UNIVERSITY

The Computer Science Department solicits applications for tenure-track and visiting faculty positions at all levels (subject to funding). Candidates for assistant professor need a PhD in computer science or computer engineering (at time of appointment) with promise for excellence in research and teaching; applicants for senior ranks must possess distinguished research records. We seek mainstream computer scientists in areas including architecture, artificial intelligence, graphics, languages and compilers, networks, software systems and engineering, and theory. In all areas we encourage applicants with interests in parallel computing, broadly defined: concurrent, distributed, vector, or systolic computing software, architectures, or applications. Salary is commensurate with rank and experience. New and visiting faculty will enjoy duties especially conducive to productive research.

The Department offers BS, MS, and PhD degrees. We have excellent cooperative research relations with industrial and government laboratories, and their people form a significant portion of our graduate student population. We operate numerous multi-user systems (HP, DEC, Sequent, ATT) and many workstations (Sun, Apple, ATT), all networked. University operations include a CYBER 205 vector supercomputer. Department personnel work in a pleasant, smoke-free environment.

Fort Collins is a growing community of 89,000 located along the foothills of the Rocky Mountains, 60 miles north of Denver. The climate is moderate—about 15 inches of precipitation and 290 days of sunshine per year. There are many cultural opportunities and year-round outdoor activities.

Send your curriculum vitae and names of at least three professional references to: Dr. R. Oldehoeft, Search Committee, Computer Science Department, Colorado State University, Fort Collins, CO 80523. Applications for January 1989 will be considered October 15, 1988. Applications for August, 1989 will be considered April 1, 1989. In each case the search will be extended if suitable candidates are not found.

Colorado State University is an EEO/AA employer. EO Office: 314 Student Services Building.

COMPUTATIONAL LINGUIST

Computational Linguist for electronic publishing firm in SE Ohio. Perform major experiments and base research to validate and extend natural language work, grammar recognition tools, sentence syntax, and discourse modeling with high emphasis on "proper noun" related research. Requires Ph.D. or ABD in Linguistics. As part of Ph.D. academic program, applicants must have taken for credit, or audited at least 5 courses in computational linguistics. Must also have presented or be preparing for presentation a Ph.D. thesis in the area of computational linguistics. It also requires a Masters in either Foreign Language or Bilingual Education or in Teaching English as a Foreign Language. Must have six months research experience. Must be able to write and speak Chinese. 40 hrs./wk., 9 am to 5 pm, \$2,670/month. Qualified applicants please reply immediately with resume to A. Mac Lean, JO# 1152210, Ohio Bureau of Employment Services, P.O. Box 1618, Columbus, Ohio, 43216.

The Artificial Intelligence Laboratory at Massachusetts Institute of Technology invites applicants to the position of VLSI Designer/Systems Programmer

Individual needed to participate in a research project leading to the design of one of the world's fastest parallel computers. Work involves register-transfer level design, logic design, digital circuit design, chip and board layout, design verification, use and maintenance of VLSI CAD tools, and maintenance and instrumentation of systems software.

Qualifications: Substantial custom VLSI design experience required. B.S. preferred.

Send resume (including names of references) to:

Ms. Marilyn Melithoniot
MIT Artificial Intelligence Laboratory
545 Technology Square, Room 807
Cambridge, MA 02140

MIT is an Affirmative Action/Equal Opportunity Employer.

DIRECTOR OF TECHNOLOGY DEVELOPMENT

Language Technology Inc. (LTI) is a leading vendor of Computer Aided Software Engineering (CASE) tools. Unlike most CASE tools, which work solely on new systems, our products work with existing systems to make them easier to understand and to extract design and documentation.

LTI is seeking a senior person to make individual contributions and provide technical leadership (the "player/coach") to a group that is developing technology that will be employed in an existing product family. The technology is in the area of program understanding and involves abstracting control and data flow from existing programs. R & D is done on Sun Workstations in Prolog.

LTI is a private, venture capital funded company located outside of Boston. Send resume to: Ms. E. Murphy, Language Technology Inc., 27 Congress St., Salem, MA 01970.

UNIVERSITY OF CALIFORNIA, DAVIS
Chair, Division of Computer Science

Nominations and applications are invited for the position of Chair, Division of Computer Science at the University of California, Davis. The chair must be a distinguished researcher with a prominent research record and an accomplished teacher. The chair is expected to provide both intellectual and administrative leadership.

The Division of Computer Science is a semi-autonomous unit within the Department of Electrical Engineering and Computer Science. The Division of Computer Science has 17 faculty members and almost 100 full time graduate students. The entire department has a total of 46 faculty members and almost 250 full time graduate students. The Division of Computer Science has experienced rapid growth over the past five years and significant further growth is expected over the next five years. The new chair is expected to have at least 7-9 new faculty positions in the next five years. Substantial startup funds are available for equipment for the chair and for new recruits. As part of the growth plans significant amounts of equipment and space have been committed to the Division of Computer Science. A new Engineering building including approximately 110K sq. ft. is scheduled for the EECS Department. In addition, a new 32K sq. ft. Engineering Research building, donated by regional industry, will be available for the research activities of the department.

Davis is a very pleasant, family-oriented community. The 900 acre, tree shaded central campus has about 20,000 students. The City of Davis enjoys the mild climate of the Sacramento Valley, and is within easy driving distance of the Sierra Nevada mountains, San Francisco, Napa Valley, and Silicon Valley. In addition, the department provides salary and benefits that are extremely attractive.

The Chair position is available July 1, 1989. Screening of nominations and applications begins October 15, 1988, and will continue until the position is filled. Please send a resume listing the names of at least three references to:

Professor Charles Martel, Chair
Search Committee
Division of Computer Science
University of California
Davis, CA 95616
Phone: (916) 752-2651

CSNET: martel%ucd.csnet@relay.cs.net
ARPANET: martel@ucdavis.edu.

The University of California, Davis, is an equal opportunity/affirmative action employer.

MICHIGAN STATE UNIVERSITY
Department of Computer Science

The Department of Computer Science invites applications for tenure track positions at all levels. Candidates from all areas of specialization in computer science or computer engineering will be considered. The department has a special interest in candidates in the areas of programming languages, artificial intelligence and expert systems, design of computer systems and networks, parallel computation, and operating systems. Candidates should have a Ph.D. in computer science or computer engineering and have a strong interest in both research and teaching.

One position is available starting in January 1989. Additional positions are anticipated beginning in September 1989. Initial screening of applications will begin on August 1, 1988. However, applications will be accepted until the positions are filled.

As a unit within the College of Engineering at Michigan State University, Computer Science offers the Bachelor of Science, Master of Science and Doctor of Philosophy degrees. Special support is available from within the college and university to initiate research by new faculty members. Faculty offices are connected to the MSUnet which provides access to an array of campus computing resources including the facilities of the College of Engineering, the Department's VAX 8600 and 64-node NCUBE, Pattern Recognition and Image Processing Laboratory, Artificial Intelligence/Knowledge Based Systems Laboratory and the Distributed Computing Research Laboratory. Access to select off-campus computers is available, as well as access to ARPANET, CSNET, BITNET and MAILNET. Additional facilities available to faculty include an IBM 3090/180E vector processor, the College's A.H. Case Center for Computer-Aided Engineering and Manufacturing, and the Electronics Research and Development Laboratory.

Michigan State University enjoys a park-like campus of 2,100 developed acres and 3,100 acres of experimental farms, outlying research facilities and natural areas. The campus is adjacent to the cities of East Lansing and the capital city, Lansing. The Greater Lansing area has approximately 250,000 residents. The communities have fine school systems and place a high value on education.

Applicants should send a resume and a statement of research and teaching interests to:

Dr. Anthony S. Wojcik, Chairperson
Department of Computer Science
A714 Wells Hall
Michigan State University
East Lansing, Michigan 48824-1027
CSNET: wojcik@cps.msu.edu

Michigan State University is an Equal Opportunity/Affirmative Action Institution and encourages applications from members of ethnic minority groups.

UNIVERSITY OF HONG KONG
Chair of Computer Science

Applications are invited for the newly-established Chair of Computer Science in the Department of Computer Science. Applicants should have strong academic qualifications and a substantial record of research and publication.

The University would prefer to make a permanent appointment, but consideration may also be given to applications for appointment on fixed or secondment terms of preferably not less than three academic years.

The University reserves the right not to fill the Chair or to fill the Chair by invitation or to make an appointment at a lower level.

Annual salary (superannuable) will be within the professional range and not less than HK\$417,240 (US\$1 = HK\$7.80 as of June 22, 1988). At current rates, salaries tax will not exceed 15½% of gross income. Housing at a charge of 7½% of salary, children's education allowances, leave, and medical benefits are provided.

Further particulars and application forms may be obtained from the Secretary General, Association of Commonwealth Universities (Appts), 36 Gordon Square, London WC1H 0PF, or from the Appointments Unit, Registry, University of Hong Kong, Hong Kong.

Closes 30 September 1988.

U.S. NAVAL ACADEMY
Tenure-Track Faculty Position
Computer Science

Possession of earned Ph.D. in computer Science or related field is required. Initial appointment is normally three years—now open. Primary responsibility will be teaching Computer Science courses. Salary commensurate with overall experience. Address inquiries to: Chairman Computer Science Department, United States Naval Academy Annapolis, MD 21402. Deadline for receipt of applications is 30 November 1988. Applications will be considered until the position is filled. An Equal Opportunity/Affirmative Action Employer.

Moving?

**PLEASE NOTIFY
US 4 WEEKS
IN ADVANCE**

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

**ATTACH
LABEL
HERE**

- This notice of address change will apply to all IEEE publications to which you subscribe.
- List new address below.
- If you have a question about your subscription, place label here and clip this form to your letter.

Name (Please Print)

New Address

City

State/Country

Zip

CONFERENCES

Editor: Edmund L. Gallizzi, Computer Science Dept., Eckerd College, St. Petersburg, FL 33733; (813) 864-8272; Compmail, e.gallizzi

Workshop to consider future transaction-processing architectures

Martin Freeman, Philips Research Laboratories, Signetics

Transactions are units of combined computational and I/O activity that are either completely done or, if not, do not change the state of the underlying system. Today, transactions are used in applications where the loss of information or the inconsistency of data across system failures is unacceptable.

Progress in semiconductor technology is now enabling transactions to be employed in more levels of our hardware/software systems and is also providing a means to simplify the algorithms and structures associated with current transaction processing systems.

The First International Workshop on Transaction Machine Architecture, to be held at Lake Arrowhead, California,

September 25-28, seeks to determine the future of transaction processing provided by this technology through survey presentations on this technology, discussions of current and future applications, panel sessions with participants making positions statements about "transactions in the year 2000," and working groups providing direction for future research.

Martin Freeman of Philips Research Laboratories, Signetics, and the outgoing chair of the Computer Society's Technical Committee on Microprocessors and Microcomputers, is the general workshop chair.

The workshop is sponsored by the TCMM as well as the society's Technical Committees on Data Engineering

and Very Large Scale Integration in cooperation with ACM's SIGMod and the UCLA Computer Science Department. It is also supported through grants from the Amdahl Corporation and the IBM Almaden Research Center.

Alan Kay of Apple Computer and Andrew Heller of IBM will present keynote speeches touching on different aspects of transactions, the user interface, and the computing environment of the future.

Dieter Gawlick of DEC heads the Program Committee.

Workshop participation is by invitation only. For further information, contact Freeman by electronic mail at mfreeman@sierra.stanford.edu, or call Doris Sung at (408) 746-3180.

Parallel processing benchmark presented at conference

*Sandra L. Bartlett and Yuval Roth-Tabak
University of Michigan at Ann Arbor*

Charles Weems of the University of Massachusetts described the Image Understanding Parallel Processing Benchmark during the Computer Society's Conference on Computer Vision and Pattern Recognition June 5-9 in Ann Arbor, Michigan.

Ramesh Jain of the University of Michigan was general conference chair, and Larry S. Davis of the University of Maryland was program chair.

Weems' paper, "IU Parallel Processing Benchmark," is the second in a series of DARPA-sponsored efforts.

The purpose of the first benchmark was to study execution times on different parallel machines for a set of 10 isolated-vision-related tasks, including Gaussian convolution, connected-component labeling, visibility of vertices in a 3D model, and subgraph isomorphism.

This second benchmark was created to study performance on a set of integrated operations that occur in a typical image-interpretation task. The test images are a set of registered intensity and range-image pairs.

To minimize the amount of programming needed to use the second benchmark, the test-image interpreting method was designed using the tasks specified for the first benchmark. While this may not be the best way to do image understanding in general or for any given parallel machine, it should help bring to light the requirements for a general vision architecture and the performance bottlenecks in different types of machines.

The hope is that this second benchmark will provide insight about the

impact parallelism can have on problems of communication and control across representations and algorithms, rather than degenerate into a simple "horse race."

Persons wishing to use the benchmark should contact Charles Weems, Computer and Information Science Department, University of Massachusetts, Amherst, MA.

Of the 260 papers submitted to the conference program committee for consideration, 134 from 15 countries were accepted covering many areas, including calibration, stereo, motion, geometry, morphology, navigation, and representation.

At the conference banquet, Dwight Carlson, a member of the governor's Entrepreneurial and Small Business Commission and president of Percepton, drew on a wealth of personal expertise as he spoke about competition in the business world today. He first pointed out that the major conflicts are no longer on the scale of Ford versus GM but, rather, the American enterprise system versus the Japanese enterprise system.

Carlson feels that major changes need to be made in the way America does business. Big business tends to become stagnant, while the National Science Foundation has shown that companies with less than 200 people are much more innovative. Small businesses need to be nurtured so their new ideas can help make America a more vital force in the world marketplace.

Government, universities, labor, and industry all have a role in this new approach to revitalizing America; individuals have a responsibility as well, he said. "Every one of these (small) companies was started by someone, and you can do it, also," Carlson pointed out.

Gil J. Ettinger of Advanced Decision Systems was presented the Roberts Prize Award sponsored by General Electric for the best conference paper. In "Large Hierarchical-Object Recognition Using Libraries of Parameterized-Model Subparts," Ettinger described an approach to develop an object-shape representation that incorporates a subpart hierarchy and scale hierarchy.

The goal is to develop a robust model-based recognition system capable of recognizing viewed objects that are scaled or mirror-image instances of the known models or that contain components with different relative scaling rotation or translation than the models. The approach should allow for efficient indexing and for automatic generation of a model library.

"YOU CAN'T DO THAT"

Build a large scale mainframe computer that will outperform the competition's leading model?

"IMPOSSIBLE!" they said.

But Amdahl did it back in the early 1970s. And today we are a leader in the development, manufacturing, marketing and support of general purpose and scientific computer systems, storage products, communications systems and software.

In less than two decades we have grown from 5 to more than 8,000 "can do" employees around the globe. Our success is a result of teamwork, innovation and commitment to achieving the impossible. If you are ready for challenge, creativity and growth, explore your opportunities with Amdahl in the following area:

Based in Silicon Valley, CA

PERFORMANCE ANALYSIS

Amdahl's breakthrough introduction of the 100 MIPS 5990 has created new frontiers in the architecture of processor performance. To meet these challenges, use your expertise in 370 architecture, mainframe operating systems or computer modeling to evaluate the performance of current and future processors. Opportunities include:

- Workload Development
- Analysis Tools
- Peripheral Performance (DASD)
- Modeling

YOU CAN join in the excitement of creating the computer systems of the future, while enjoying the benefits and competitive salary you would expect from an industry leader.

YOU CAN send your resume to B. Coburn at Amdahl Corporation, Employment Department 8-17, P.O. Box 3470, M/S 300, Sunnyvale, California 94088-3470. Principals only, please.

Amdahl Corporation is proud to be an equal opportunity employer through affirmative action.

YOU CAN AT
amdahl

CALENDAR

 Conferences that the Computer Society sponsors or participates in are indicated by the Computer Society logo; additional conference sponsors are also listed. Other conferences of interest to our readers are also included.

For inclusion in Call for Papers or Calendar, submit information six weeks before the month of publication (i.e., for the November 1988 issue, send information for receipt by Sept. 15, 1988) to Calendar Editor, *Computer*, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720.

September 1988

Mapping and Geographic Information Systems 88, Sept. 12-15, Orlando, Fla. Sponsor: National Computer Graphics Assoc. Contact NCGA, 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031.

Eurographics 88, Ninth European Assoc. for Computer Graphics Conference, Sept. 12-16, Nice, France. Sponsor: INRIA. Contact INRIA, Service des Relations Exterieures, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, France, phone 33 (1) 39-63-55-01.

 **Workshop on Future Trends of Distributed Computing Systems in the 1990s, Sept. 14-16, Hong Kong.** Contact Tien Chi Chen, United College, Chinese University of Hong Kong, Shatin, N.T., Hong Kong, phone 852 (5) 695-2576.

 **Fourth International Conference of Modeling Techniques and Tools for Computer Performance Evaluation, Sept. 15-17, Palma de Mallorca, Spain.** Other sponsors: UIB, ATI. Contact Ramon Puigjaner, UIB, Miguel dels Sants Oliver 2, 07012 Palma de Mallorca, Spain, phone 34 (71) 72-57-06.

 **IEEE Artificial Neural Networks Conference, Sept. 18-21, Reston, Va.** Contact Kamal Karma, 823 Flegler Rd., Gaithersburg, MD 20879, phone (301) 984-7657.

Fourth International Symposium on Biological and Artificial Intelligence Systems, Sept. 18-22, Trento, Italy. Contact Alberta Martino, IBM Corp., Dept. 48B/428, Neighborhood Rd., Kingston, NY 12401, phone (914) 385-4964.

Sixth Software Quality Conference, Sept. 19-20, Portland, Ore. Sponsor: Pacific Northwest Software Quality Conference Committee. Contact Mary Lawrence, Lawrence & Craig, 320 SW Stark, Rm. 411, Portland, OR 97204, phone (503) 222-2606.

Conference on Computerized Assistance During System Life Cycle, Sept. 19-22, London. Sponsor: International Federation for Information Processing. Contact IFIP, 3 rue de Marche, CH-1204, Geneva, Switzerland.

Workshop and Symposium on Formal Techniques in Real-time and Fault-Tolerant Systems, Sept. 20-23, Coventry, England. Sponsor: University of Warwick. Contact M. Joseph, Dept. of Computer Science, University of Warwick, Coventry, CV4 7AL, UK.

 **First International Workshop on Transaction Machine Architecture, Sept. 25-28, Lake Arrowhead, Calif.** Contact Martin Freeman, Signetics Corp., 811 E. Arques Ave., Sunnyvale, CA 94086, phone (408) 991-3591.

OOPSLA 88, Third Conference on Object-Oriented Programming: Systems, Languages and Applications, Sept. 25-30, San Diego, Calif. Sponsor: ACM. Contact Barbara Noparstak, Digitalk, Inc., 9841 Airport Blvd, Los Angeles, CA 90045, phone (213) 645-1082.

CSCW 88, Second Conference on Computer-Supported Cooperative Work, Sept. 26-28, Portland, Ore. Sponsor: ACM. Contact Julie Kling, Lotus Development, 55 Cambridge Pkwy., Cambridge, MA 02142, phone (617) 577-8500.

 **Computational Intelligence 88, Sept. 26-30, Milan, Italy.** Other sponsors: ACM Italian chapter, University of Milan. Contact Giorgio Valle, Università di Milano, Dip. di Scienze dell'Informazione, Via Moretto 9, 20133 Milano, Italy, phone 39 (2) 27-72-228.

 **Second International Workshop on Object-Oriented Programming Database Systems, Sept. 27-30, Bad Muenster am Stein-Ebernburg, Germany.** Other sponsors: Gesellschaft für Informatik, FZI. Contact Alex Buchmann, CCA, 4 Cambridge Center, Cambridge, MA 02142, phone (617) 492-8860.

26th Allerton Conference on Communication, Control, and Computing, Sept. 28-30, Monticello, Ill. Sponsor: University of Illinois at Urbana-Champaign. Contact Allerton Conference, c/o Mark W. Spong, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1101 W. Springfield Ave., Urbana, IL 61801, phone (217) 333-4281.

October 1988

 **ICCD 88, IEEE International Conference on Computer Design, Oct. 2-6, Rye Brook, N.Y.** Contact Prathima Agrawal, ICCD 88, AT&T Bell Laboratories, 600 Mountain Ave., Rm. 3D-480, Murray Hill, NJ 07974, phone (201) 582-6943.

EWSL 88, Third European Working Session on Learning, Oct. 3-5, Glasgow, Scotland. Contact Derek Sleeman, Dept. of Computing Science, University of Aberdeen, Aberdeen AB9 2UB, Scotland, UK, phone 44 (224) 27-22-88.

 **Compsac 88, 12th International Computer Software and Applications Conference, Oct. 3-7, Chicago.** Contact Wing N. Toy, Compsac 88, AT&T Bell Laboratories, Rm. 1Z-306, 200 Park Plaza, Naperville, IL 60566, phone (312) 416-4046.

1988 International Display Research Conference, Oct. 4-6, San Diego, Calif. Sponsors: IEEE, Society for Information Display. Contact Palisades Institute for Research Services, Attn. IDRC, 201 Varick St., New York, NY 10014, phone (212) 620-3388.

Northcon 88, Oct. 4-6, Seattle. Sponsors: IEEE, Electronics Representatives Assoc., Electronics Manufacturers Assoc. Contact Electronic Conventions Management, 8110 Airport Blvd., Los Angeles, CA 90045-3194, phone (213) 772-2965.

 **International Workshop on Defect and Fault Tolerance in VLSI Systems, Oct. 6-7, Springfield, Mass.** Contact Israel Koren, Dept. of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003, phone (413) 545-2643.

NCGA CAD/CAM 88, Oct. 9-12, Boston. Sponsor: National Computer Graphics Assoc. Contact NCGA CAD/CAM 88, 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031.

 **ICCL 88, International Conference on Computer Languages, Oct. 9-13, Miami Beach, Fla.** Contact Pei Hsia, Computer Science Engineering Dept., University of Texas, 2100 Oak Bluff Dr., Arlington, TX 76019, phone (817) 273-3610.

Seventh Symposium on Reliable Distributed Systems, Oct. 10-12, Columbus, Ohio. Contact David Cohen, Tekuekron Infoswitch, 1784 Firman Dr., Richardson, TX 75081; or Ming T. Liu, Dept. of Computer and Information Science, Ohio State University, 2036 Neil Ave., Columbus, OH 43210-1277, phone (614) 292-1837.

1988 IEEE Workshop on Visual Languages, Oct. 10-12, Pittsburgh. Cosponsor: University of Pittsburgh. Contact A.T. Bertziss, Dept. of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260.

13th Conference on Local Computer Networks, Oct. 10-12, Minneapolis, Minn. Contact Ron Rutledge, Martin Marietta Energy Systems, Bldg. 4500N, MS-271, PO Box X, Oak Ridge National Laboratory, Oak Ridge, TN 37831, phone (615) 576-7643.

Frontiers 88, Second Symposium on the Frontiers of Massively Parallel Computation, Oct. 10-12, Fairfax, Va. Cosponsor: George Mason University, NASA. Contact James R. Fischer, Frontiers 88 Symposium, Code 635, NASA Goddard Space Flight Center, Greenbelt, MD 20771, phone (301) 286-3464.

International Workshop on Computer Vision, Oct. 12-14, Tokyo. Sponsor: International Assoc. for Pattern Recognition. Contact Mikio Takagi, Institute of Industrial Science, University of Tokyo, 7-22-1, Roppongi, Minato-ku, Tokyo 106, Japan, phone 81 (3) 479-0289.

SIGGraph Symposium on User Interface Software, Oct. 17-19, Banff, Canada. Sponsor: ACM. Contact John Sibert, Dept. of Electrical Engineering and Computer Science, George Washington University, Washington, DC 20052, phone (202) 994-4953.

Fourth International Conference on Satellite Systems for Mobile Communications and Navigation, Oct. 17-19, London. Sponsor: Institution of Electrical Engineers. Contact Conference Services Dept., IEE, Savoy Pl., London WC2R 0BL, UK, phone 44 (1) 240-1871.

Third International Symposium on Knowledge Engineering, Oct. 17-21, Madrid, Spain. Sponsor: American Assoc. for Artificial Intelligence. Contact Jose R. Chelala, Third International Symposium on Knowledge Engineering, Alvarez de Baena, 3-2, 28006 Madrid, Spain.

Fourth Software Engineering Conference and Exhibition, Oct. 18-21, Paris. Sponsor: AFCET. Contact CGL4, 156 blvd. Pereire, 75017 Paris, France.

Concom 88, Advances in Communications and Control Systems, Oct. 19-20, Baton Rouge, La. Contact Morteza Naraghi-Pour, Dept. of Electrical and Computer Engineer-

ing, Louisiana State University, Baton Rouge, LA 70803, phone (504) 388-5551.

FOCS, 88, 29th Foundations of Computer Science Conference, Oct. 24-26, White Plains, N.Y. Contact Christos Papadimitrou, Dept. of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92043, phone (619) 534-2086.

CSM 88, Conference on Software Maintenance, Oct. 24-27, Phoenix, Ariz. Contact Robert Arnold, Software Productivity Consortium, 1880 Campus Commons Dr. North, Reston, VA 22091, phone (703) 391-1898.

First International Symposium on Artificial Intelligence: Knowledge-Based Systems, Oct. 24-28, Monterrey N.L., Mexico. Contact Francisco J. Cantu, Instituto Tecnológico de Monterrey, ITESM Suc. Correos "J", Monterrey N.L., Mexico, 64849, phone 52 (83) 59-59-43.

Second International Software for Strategic Systems Conference, Oct. 25-26, Huntsville, Ala. Cosponsor: University of Alabama in Huntsville. Contact Karen B. Mack, University of Alabama in Huntsville, Division of Continuing Education, Tom Bevill Center, Huntsville, AL 35899, phone (205) 895-6357.

IECON 88, 14th Conference of the IEEE Industrial Electronics Society, Oct. 25-27, Singapore. Contact V.K.L. Huang, AT&T Information Systems, Crawford Corners Rd., Holmdel, NJ 07733, phone (210) 949-0069.

Systec 88, Second International Trade Fair for Computer-Integrated Manufacturing, Oct. 25-28, Munich, West Germany. Contact Munchener Messe- und Ausstellungsgesellschaft mbH, Projektleitung Systec 88, Postfach 12 10 09, D-8000 München 12, West Germany, phone 49 (89) 51-070.

Ninth IEEE Symposium on Mass Storage Systems, Oct. 30-Nov. 3, Monterey, Calif. Contact Patric Savage, Shell Development Co., PO Box 481, Houston, TX 77001, phone (713) 663-2384.

ICCC 88, Ninth International Conference on Computer Communication, Oct. 30-Nov. 3, Tel Aviv, Israel. Sponsor: IEEE Israel Section. Contact J. Kella, Secretariat, PO Box 50006, Tel Aviv 61500, Israel, phone 972 (3) 654-571.

22nd Asilomar Conference on Signals, Systems, and Computers, Oct. 31-Nov. 2, Pacific Grove, Calif. Cosponsor: Naval Postgraduate School. Contact Douglas Elliott, Rockwell International, 3370 Miraloma Ave., Mail Stop BD07, Anaheim, CA 92803-317, phone (714) 762-2340.

ICCS 88, International Conference on Communication Systems, Oct. 31-Nov. 3, Singapore. Sponsors: IEEE, National University of Singapore, Telecommunication Authority

of Singapore. Contact ICCS 88, c/o Meeting Planners Pte Ltd., 100 Beach Rd., #33-04, Shaw Towers, Singapore 0718.

November 1988

1988 Workshop on VLSI Signal Processing, Nov. 2-4, Monterey, Calif. Sponsor: IEEE. Contact Paulette Powell, Electronics Research Laboratory, University of California, Berkeley, CA 94720, phone (415) 642-1566.

Symposium on Computer Graphics Education, Nov. 4-5, Poughkeepsie, N.Y. Sponsors: Marist College, ACM. Contact William J. Joel, Marist College, 82 North Rd., Poughkeepsie, NY 12601.

12th Symposium on Computer Applications in Medical Care, Nov. 6-9, Washington, DC. Contact Office of Continuing Medical Education, George Washington University Medical Center, 2300 K St. NW, Washington, DC 20037, phone (202) 994-8928.

ICCAD 88, IEEE International Conference on Computer-Aided Design, Nov. 7-10, Santa Clara, Calif. Cosponsor: ACM. Contact Al Jimenez, Mentor Graphics Corp., 1940 Zanker Rd., San Jose, CA 95112, phone (408) 436-1500; or Pro CASE, 3130 Dela Cruz Blvd., Suite 100, Santa Clara, CA 95054, phone (408) 727-0714.

Third Workshop on Knowledge Acquisition for Knowledge-Based Systems, Nov. 7-11, Banff, Canada. Sponsor: American Assoc. for Artificial Intelligence. Contact John H. Boose, Advanced Technology Center, Boeing Computer Services, MS 7L-64, PO Box 24346, Seattle, WA 98124, phone (206) 865-3253.

GOMAC 88, Government Microcircuit Applications Conference, Nov. 8-10, Las Vegas, Nev. Contact C. Edward Holland, Jr., Semiconductor Research Corp., 4501 Alexander Dr., PO Box 12053, Research Triangle Park, NC 27709, phone (919) 541-9400.

Symposiums on Nuclear Science and Nuclear Power Systems, Nov. 9-11, Orlando, Fla. Sponsor: IEEE. Contact E. Barsotti, Fermilab, MS 222, Box 500, Batavia, IL 60510, phone (312) 840-4061.

Second International Symposium on Interoperable Information Systems, Nov. 10-11, Tokyo. Cosponsor: INTAP. Contact Hideo Aiso, Dept. of Electrical Engineering, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Karagawa, 223 Japan, phone 81 (44) 63-1141, ext. 3320.

Supercomputing 88, Nov. 14-18, Kissimmee, Fla. Cosponsor: ACM. Contact George Michael, Lawrence Livermore National Laboratory, PO Box 808, L-306, Livermore, CA 94550, phone (415) 422-4239.

Ninth International Conference on Pattern Recognition, Nov. 14-17, Rome. Sponsor: International Assoc. for Pattern Recognition. Contact M.J.B. Duff, Dept. of Physics and Astronomy, University College London, Gower St., London WC1E 6BT, England, UK, phone 44 (1) 380-7010; Herbert Freeman, CAIP Center, Rutgers University, Piscataway, NJ 08855-1390, phone (201) 932-3443; or Stefano Levialdi, Dipartimento di Matematica, Università di Roma, Piazzale Aldo Moro 2, I-00185 Roma, Italy, phone 39 (6) 49-91-32-49 or 50.

Seventh International Conference on Entity-Relationship Approach, Nov. 16-18, Rome. Cosponsor: ER Institute. Contact Salvatore T. March, Management Sciences, University of Minnesota, 271 19th Ave. South, Minneapolis, MN 55455, phone (612) 624-2017.

Fourth Conference on Artificial Intelligence for Space Applications, Nov. 15-16, Huntsville, Ala. Sponsors: NASA, University of Alabama in Huntsville. Contact Conference on Artificial Intelligence for Space Applications, NASA/EB44, MSFC, AL 35812, phone (205) 544-5181.

AIDA 88, Fourth Conference on Artificial Intelligence and Ada, Nov. 15-16, Washington, DC. Sponsor: George Mason University. Contact Jorge Diaz-Herrera, Computer Science Dept., George Mason University, 400 University Dr., Fairfax, VA 22030, phone (703) 323-2713.

Sixth CIPS Edmonton Computer Conference, Nov. 15-17, Edmonton, Canada. Sponsor: Canadian Information Processing Society. Contact U.M. Maydell, Dept. of Computing Science, University of Alberta, Edmonton, Alberta, T6G 2H1, Canada.

Wescon 88, Nov. 15-17, Anaheim, Calif. Sponsor: IEEE. Electronic Representatives Assoc. Contact Wescon 88, c/o Electronic Conventions Management Educational Activities Dept., 8110 Airport Blvd., Los Angeles, CA 90045, phone (213) 772-2965.

12th Western Educational Computing Conference, Nov. 17-18, San Diego, Calif. Sponsor: California Educational Computing Consortium. Contact Judah Rosenwald, Extended Education, NAD 153, San Francisco State University, 1600 Holloway, San Francisco, CA 94132, phone (415) 338-1212.

Micro 21, Workshop on Microprogramming and Microarchitecture, Nov. 28-Dec. 1, San Diego, Calif. Cosponsor: ACM. Contact Wen-Mei Hwu, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1101 W. Springfield Ave., Urbana, IL 61801, phone (217) 244-8270.

Conference on Neural Information Processing Systems, Nov. 28-Dec. 1, Denver. Sponsor: IEEE. Contact Scott Kirkpatrick, IBM T.J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598.

FGCS 88, International Conference on Fifth-Generation Computer Systems, Nov. 28-Dec. 2, Tokyo. Cosponsor: Institute for New Generation Computer Technology. Contact Hideo Aiso, Dept. of Electrical Engineering, Keio University, 3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Karagawa, 223 Japan, phone 81 (44) 63-1141, ext. 3320.

December 1988

Globecom 88, Dec. 1-2, Hollywood, Fla. Sponsor: IEEE. Contact Dennis J. Sassa, Bell Communications Research, NVC 2Z-129, 331 Newman Springs Rd., Red Bank, NJ 07701-7020, phone (201) 446-6787.

International Symposium on Databases for Parallel and Distributed Systems, Dec. 5-7, Austin, Texas. Contact Joseph E. Urban, Dept. of ECE, University of Miami, PO Box 248294, Coral Gables, FL 33124, phone (305) 284-3598; or Won Kim, MCC, 3500 Balcones Center Dr., Austin, TX 78759, phone (512) 338-3439.

SIGSoft 88, Third Symposium on Software Development Environments, Dec. 5-7, Boston. Sponsor: ACM. Contact Barry Boehm, TRW-DSG, 1 Space Park, R2-2086, Redondo Beach, CA 90278.

Ninth Real-Time Systems Symposium, Dec. 6-8, Huntsville, Ala. Contact Walter L. Heimerdinger, Honeywell, Systems and Research Center, MN65-2100, 3660 Technology Dr., Minneapolis, MN 55418, phone (612) 782-7319.

EuroComm 88, Second International Exhibition and Congress on Business, Public, and Home Communications, Dec. 6-9, Amsterdam. Contact H. Heringa, EuroComm 88, c/o RAI Gebouw bv, Europaplein, 1078 GZ Amsterdam, The Netherlands, phone 31 (20) 549-1212.

International Seminar on Performance of Distributed and Parallel Systems, Dec. 7-9, Kyoto, Japan. Contact Yutaka Takahashi, Dept. of Applied Mathematics and Physics, Kyoto University, Kyoto 606, Japan, phone 81 (75) 751-2111, ext. 5513.

Winter Simulation Conference, Dec. 12-14, San Diego, Calif. Contact Peter Haigh, NCR Corp., 1700 S. Patterson Blvd., SER Bldg., Dayton, OH 45479, phone (513) 865-8042.

ICCV 88, Second International Conference on Computer Vision, Dec. 12-15, Tarpon Springs, Fla. Contact Ruzena Bajcsy, Computer and Information Science Dept., University of Pennsylvania, 200 S. 33rd St., Philadelphia, PA 19104-6389, phone (215) 898-6222.

Fourth Aerospace Computer Security Applications Conference, Dec. 12-16, Orlando, Fla. Cosponsors: American Society

for Information Science, Aerospace Computer Security Associates. Contact Marshall Abrams, 1820 Dolley Madison Blvd., McLean, VA 22101, phone (703) 883-6938; or Robert D. Kovach, ACSA, c/o ORI, 1375 Piccard Dr., Rockville, MD 20850.

Second International Workshop of VLSI Design, Dec. 15-18, Bangalore, India. Sponsor: Computer Society of India. Contact Ravi M. Apte, Valid Logic Systems, 2820 Orchard Pkwy, San Jose, CA 95134, phone (408) 432-9400; or A. Prabhakar, Indian Telephone Industries, Dooravani Nagar, Bangalore 560 016, India, phone 91 (812) 563-211.

International Computer Science Conference, Dec. 19-21, Hong Kong. Contact Jean-Louis Lassez, IBM T.J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598; or Francis Y.L. Chin, Center of Computer Studies and Application, University of Hong Kong, Hong Kong.

ICS 88, International Computer Symposium, Dec. 20-21, Taipei, Taiwan. Contact Louis R. Chow, Tamkang University, Tamsui, Taiwan, Republic of China; or Shi-Kuo Chang, Dept. of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260.

Eighth Conference on Foundations of Software Technology and Theoretical Computer Science, Dec. 21-23, Pune, India. Contact K.V. Nori, TRDDC, 1 Mangaldas Rd., Pune 411001, India, phone 91 (212) 61-608.

10th Israel Convention on CAD/CAM and Robotics, Dec. 27-29, Tel Aviv. Sponsor: Israel Society for Computer-Aided Design and Manufacturing. Contact Secretariat, Ortra, Ltd., PO Box 50432, Tel Aviv 61500, Israel, phone 972 (3) 66-48-25.

January 1989

International Conference on Wafer Scale Integration, Jan. 3-5, San Francisco. Contact Earl Swartzlander, TRW, R2/2076, 1 Space Park, Redondo Beach, CA 90278, phone (213) 812-0791.

22nd Hawaii International Conference on Systems Sciences, Jan. 3-6, Kailua-Kona, Hawaii. Contact Ralph Sprague, Jr., Decision Sciences Dept., University of Hawaii, 2404 Maile Way, E-303, Honolulu, HI 96822, phone (808) 948-7430.

Impact of Recent Computer Advances on Operations Research, Jan. 4-6, Williamsburg, Va. Sponsor: Operations Research Society of America. Contact Ramesh Sharda, College of Business Administration, Oklahoma State University, Stillwater, OK 74078, phone (405) 624-5113.

Second International Workshop on Artificial Intelligence in Economics and Management, Jan. 11-13, Singapore. Cosponsors: International Federation for

Information Processing, International Federation of Operations Research Societies, International Federation of Automatic Control. Contact Juzar Motiwalla or Vicky Toh, Institute of Systems Science, National University of Singapore, Heng Mui Keng Terrace, Singapore 0511, phone (65) 772-2075.

February 1989

Aerospace Applications Conference, Feb. 5-10, Breckenridge, Colo. Sponsor: IEEE. Contact Leo Mallette, Hughes Aircraft, MS: Bldg. R-10, A9026, PO Box 92919, Los Angeles 90009, phone (213) 334-2909.

 **Fifth International Conference on Data Engineering Feb. 7-9, Los Angeles.** Contact John Carlis, Computer Science Dept., University of Minnesota, 207 Church St., SE, Minneapolis, MN 55455, phone (612) 625-6092; or Data Engineering 89, Computer Society, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202) 371-1013.

17th Computer Science Conference, Feb. 21-23, Louisville, Ky. Sponsor: ACM. Contact Conference Dept. A, Assoc. for Computing Machinery, 11 W. 42nd St., New York, NY 10036.

 **Compcon Spring 89, Feb. 27-Mar. 3, San Francisco.** Contact Kenichi Miura, Computational Research Dept., MS B2-7, Fujitsu America, 3055 Orchard Dr., San Jose, CA 95134-2017, phone (409) 432-1300, ext. 5408 or 5723; or Compcon Spring 89, Computer Society, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202) 371-1013.

March 1989

Fourth Conference on Hypercube Concurrent Computers and Applications, Mar. 6-8, Monterey, Calif. Contact John Gustafson, Sandia National Laboratories, Division 3538, Bldg. MO-196, Albuquerque, NM 87185.

EMC 89, Eighth Zurich Symposium and Technical Exhibition on Electromagnetic Compatibility, Mar. 6-9, Zurich. Sponsor: Swiss Electrotechnical Assoc. Contact T. Dvorak, EMC 89, ETH Zentrum-IKT, CH-8092, Zurich, Switzerland, phone 41 (1) 256-2790.

 **CAIA 89, Fifth International Conference on Artificial Intelligence Applications, Mar. 6-10, Miami, Fla.** Contact CAIA 89, Computer Society, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202) 371-1013.

Seventh National Conference on Ada Technology, Mar. 13-16, Atlantic City, NJ. Contact Seventh National Conference on Ada

IEEE GLOBECOM '88 Workshop on:

THE APPLICATION OF EMERGING TECHNOLOGIES IN NETWORK OPERATIONS AND MANAGEMENT

Where: Diplomat Hotel, Hollywood, Florida
When: December 1-2, 1988
Sponsors: IEEE Communications Society, and the Committee on Network Operations and Management
Target Audience: Network operations planners, system engineers, developers, and managers
Purpose: To explore the introduction of AI, expert systems, object models, database access, speech, and the integration of these technologies into network operations. Follows up on the ICC'87 workshop on expert systems in network operations. Will cover both successful applications as well as some applications that have not fulfilled their promise.

AGENDA

December 1, AM

Tutorials on object modeling, database access, and expert systems

December 1, PM

December 1, 7:00 PM

Case studies of selected applications
Dinner and keynote address by Larry Bernstein, Executive Director, AT&T Bell Laboratories

December 2, AM

Papers and panel discussion on the future of these applications

RESERVATIONS

Fee:

\$300 (includes workshop binder, continental breakfast and lunch both days, dinner December 1)

Deadline for reservations:

October 1, 1988
(limited to first 150 received)

Contact:

Dennis J. Sassa,
Bell Communications Research
NVC 2Z-129
331 Newman Springs Rd.
Red Bank, NJ 07701-7020
(201)446-6787
(for hotel reservations
contact Globecom '88)

CALL FOR PAPERS

You are cordially invited to submit case studies or other papers for presentation at the workshop; send an abstract and paper or series of viewgraphs to Dennis J. Sassa at the above address. Papers must be received before August 15, 1988.



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.



Technology, US Army Communications—Electronics Command, Attn.: AMSEL-RD-SE-CRM (Kay Trezza), Fort Monmouth, NJ 07703-5000, phone (201) 532-1898.

Workshop on Visual Motion, Mar. 20-22, Irvine, Calif. Contact Ellen Hildreth, Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139; or Ramesh Jain, Electrical Engineering and Computer Science Dept., University of Michigan, Ann Arbor, MI 48109-2122.

PCCC 89, Phoenix Conference on Computers and Communications, Mar. 22-24, Scottsdale, Ariz. Contact Thaddeus Regulinski, Loral Corp., PO Box 295, Goodyear, AZ 85338.

AISIG 89, Artificial Intelligence Systems in Government Conference, Mar. 27-31, Washington, DC. Contact AISIG 89, MS W418, Mitre Corp., 7525 Colshire Dr., McLean, VA 22102; or Computer Society, 1730 Massachusetts Ave. NW, 20036-1903, phone (202) 371-1013.

Third Parallel Processing Symposium, Mar. 29-31, Fullerton, Calif. Cosponsors: Orange County Computer Society Chapter, California State University, Fullerton. Contact Larry Canter, 1619 N. Hale, Fullerton, CA 92631, phone (714) 738-3414.

Eighth SIGAct-SOGMod-SIGArt Symposium on Principles of Database Systems, Mar. 29-31, Philadelphia. Sponsor: ACM. Contact Avi Silbershatz, Computer Science Dept., University of Texas at Austin, Austin, TX 78712.

Workshop on Applied Computing 89, Mar. 30-31, Stillwater, Okla. Contact Donald D. Fisher, Oklahoma State University, CIS, Stillwater, OK 74078, phone (405) 624-5668.

CMC 89, Colorado Microelectronics Conference, Mar. 30-31, Colorado Springs, Colo. Contact Conference Secretary, CMC 89, Microelectronics Research Laboratories, University of Colorado, Box 7150, Colorado Springs, CO 90933-7150, phone (719) 593-3488.

Western Educational Computing Conference, Mar. 30-31, Santa Cruz, Calif. Sponsor: California Educational Computing Consortium. Contact Judah Rosenwald, Extended Education, NAD 153, San Francisco State University, 1600 Holloway, San Francisco, CA 94132, phone (415) 338-1212

April 1989

Second National Conference on Telecommunications, Apr. 2-5, York, England. Sponsor: Institution of Electrical Engineers. Contact IEE Conference Services, Savoy Pl., London WC2R 0BL, UK, phone 44 (1) 240-1871, ext. 222.

ASPLOS III, Third International Conference on Architectural Support for

Programming Languages and Operating Systems, Apr. 3-6, Boston. Cosponsor: ACM. Contact Joel Emer, DEC/MIT, 545 Technology Square (NE43-503), Cambridge, MA 02139.

Working Conference on Visual Database Systems, Apr. 3-7, Tokyo. Sponsors: IFIP IPSJ. Contact IFIP TC-2 Working Conference, Tosiya L. Kunii, Dept. of Information Science, Faculty of Science, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan, phone 81 (3) 812-2111, ext. 4116.

International Symposium on Database Systems for Advanced Applications, Apr. 10-12, Seoul, Korea. Sponsors: IPSJ, KISS. Contact Sukho Lee, Dept. of Computer Engineering, Seoul National University, Sinlim-Dong, Gwanak-Ku, Seoul 151, Korea, phone 82 (2) 886-0101.

MIV 89, International Workshop on Industrial Applications of Machine Intelligence and Vision, Apr. 10-12, Tokyo. Sponsor: IEEE. Contact Mitsuru Ishizuka, Institute of Industrial Science, University of Tokyo, 7-22-1, Roppongi, Minato-ku, Tokyo, 106, Japan, phone 81 (03) 470-5389.

ETC 89, First European Test Conference, Apr. 12-14, Paris. Sponsor: SEE. Contact Colin Maunder, British Telecom Research Laboratories, Martlesham Heath, Ipswich, Suffolk IP5 7RE, phone 44 (473) 642-706.

Multimedia 89, Second Comsoc International Multimedia Communications Workshop, Apr. 20-23, Ottawa, Canada. Sponsor: IEEE. Contact Ottawa Carleton Research Institute, 1150 Morrison Dr., Suite 302, Ottawa, Ontario, Canada K2H 8S9, phone (613) 726-8827.

Infocom 89, Conference on Computer Communications, Apr. 24-27, Ottawa, Canada. Contact J.W. Mark, IEEE Infocom 89, Dept. of Electrical Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1, phone (519) 888-4016.

Vision 89, Apr. 25-27, Chicago. Sponsor: Society of Manufacturing Engineers. Contact Maria Nowakowski, SME, 1 SME Dr., PO Box 930, Dearborn, MI 48121, phone (313) 271-1500, ext. 376.

CHI 89, Conference on Human Factors in Computing Systems, Apr. 30-May 4, Austin, Texas. Cosponsors: ACM, HFS. Contact Claudia Raun or Bill Curtis, MCC, PO Box 200195, Austin, TX 78720, phone (512) 338-3798.

34th International Instrumentation Symposium, Apr. 30-May 4, Orlando, Fla. Sponsor: ISA. Contact Frederick A. Kern, PO Box 65, Seaford, VA 23696, phone (804) 865-3269.

May 1989

1989 IEEE Symposium on Research in Security and Privacy, May 1-3, Oak-

land, Calif. Contact Terry V. Benz, Trusted Information Systems, 11340 Olympic Blvd., Suite 265, Los Angeles, CA 90064, phone (213) 477-5828.

Sixth Canadian Symposium on Instructional Technology, May 3-5, Halifax, Nova Scotia, Canada. Sponsor: National Research Council Canada. Contact Laurier Forget, Sixth Canadian Symposium on Instructional Technology, Conference Services, NRCC, Ottawa, Ontario K1A 0R6, Canada, phone (613) 993-9009.

CompEuro 89, International Conference on VLSI and Computer Peripherals, May 8-12, Hamburg. Contact Walter E. Proebster, IBM Laboratory, PO Box 1380, D-7030 Boeblingen, Federal Republic of Germany, phone 49 (70) 3116-3929.

ICCAL 89, Second International Conference on Computer-Assisted Learning, May 9-11, Dallas. Sponsor: Computer Learning Research Center. Contact Janet Harris, Center for Continuing Education, University of Texas at Dallas, PO Box 830688, MS CN 1.1, Richardson, TX 75083-0688.

CCC 89, Second Hungarian Custom Circuit Conference, May 10-12, Szeged, Hungary. Sponsor: MATE. Contact MATE Secretariat, 1055 Budapest, Kossuth L. tér 6-8, Hungary, phone (1) 531-406.

International Conference on Robotics and Automation, May 14-19, Scottsdale, Ariz. Sponsor: IEEE. Contact Harry Hayman, PO Box 3216, Silver Spring, MD 20901, phone (301) 434-1990 or (407) 483-3037.

11th International Conference on Software Engineering, May 15-18, Pittsburgh. Cosponsor: ACM. Contact Larry Druffel, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890, phone (412) 268-7740.

Fifth International Workshop on Software Specification and Design, May 19-20, Pittsburgh. Contact Colin Potts, MCC, 9390 Research Blvd., Kaleido II Bldg., Austin, TX 78759, phone (512) 338-3629.

SIGMetrics and Performance 89, May 23-26, Berkeley, Calif. Sponsors: ACM, IFIP. Contact Luis-Felipe Cabrera, IBM Almaden Research Center, Mail Code K52/803, San Jose, CA 95120-6099, phone (408) 927-1838.

16th International Symposium on Computer Architecture, May 28-June 1, Jerusalem, Israel. Cosponsor: ACM. Contact 16th International Symposium on Computer Architecture, PO Box 3190, Tel Aviv 61031, Israel, phone 972 (3) 246-261.

19th International Symposium on Multiple-Valued Logic, May 29-31, Guangzhou, China. Contact David M. Miller, Dept. of Computer Science, University of Victoria, PO Box 1700, Victoria, B.C., Canada V8W 2Y2, phone (604) 721-7220.



THE COMPUTER SOCIETY
OF THE IEEE

Membership Application



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

HOW TO JOIN

Membership dues and publication subscriptions are annualized to December 31.
Pay the full-year fee if application is mailed September 1–February 29.
Pay the half-year fee if application is mailed March 1–August 31.

- 1 COMPUTER SOCIETY ONLY** (affiliate membership).
You are eligible if you are seriously interested in any aspect of the computer field or if you are a member of one of the affiliate societies listed below.
(Check all applicable societies.)

Half-Year Mar 1-Aug 31 Full-Year Sept 1-Feb 29

Affiliate Societies

☐ \$19.50 ☐ \$39.00

- | | |
|--|---|
| ☐ American Institute of Aeronautics and Astronautics (AIAA) | ☐ Instrument Society of America (ISA) |
| ☐ American Institute of Physics (AIP) | ☐ Mathematical Association of America |
| ☐ American Mathematical Society (AMS) | ☐ National Association of Accountants |
| ☐ American Society of Mechanical Engineers | ☐ Operations Research Society of America (ORSA) |
| ☐ Association for Computing Machinery (ACM) | ☐ Society of Aircraft Materials and Processes (SAMPE) |
| ☐ Australian Computer Society | ☐ Society of Automotive Engineers (SAE) |
| ☐ British Computer Society | ☐ Society for Industrial and Applied Mathematics (SIAM) |
| ☐ Data Processing Management Association (DPMA) | ☐ Society for Computer Simulation (SCS) |
| ☐ Information Processing Society of Japan (IPSJ) | ☐ Society for Photo-Optical Instrumentation Engineers (SPIE) |
| ☐ Institute of Electronic, Information and Communication Engineers (IEICE) of Japan | ☐ The Computer Society of the Republic of China (Taiwan) |
| ☐ Institution of Electrical Engineers (IEE-UK) | |

- 3 COMPUTER SOCIETY MEMBERSHIP FOR IEEE MEMBERS.** If you are presently an IEEE member, you may join the Computer Society for a nominal amount. (Complete only shaded area of application.)

Half-Year Mar 1-Aug 31 Full-Year Sept 1-Feb 29

IEEE Member Number _____

☐ \$ 7.50 ☐ \$15.00

OPTIONAL PUBLICATIONS

- 4** Whichever membership option you chose, you are now eligible to subscribe at low member rates to these periodicals.

	(issues per year)	Half-Year Mar 1-Aug 31	Full-Year Sept 1-Feb 29
IEEE Computer Graphics & Applications (3061)...	6	☐ \$ 9.50	☐ \$19.00
IEEE Design and Test (3111)	6	☐ \$10.00	☐ \$20.00
IEEE Expert (3151)	4	☐ \$ 6.00	☐ \$12.00
IEEE Micro (3071)	6	☐ \$ 9.00	☐ \$18.00
IEEE Software (3121)	6	☐ \$ 8.50	☐ \$17.00
Transactions on Computers (1161)	12	☐ \$ 9.00	☐ \$18.00
Transactions on Pattern Analysis and Machine Intelligence (1351)	6	☐ \$ 7.50	☐ \$15.00
Transactions on Software Engineering (1171)	12	☐ \$ 9.00	☐ \$18.00

- 2 COMPUTER SOCIETY AND IEEE.*** In addition to your Computer Society benefits, you'll receive many IEEE privileges and benefits. You are eligible if your technical interests are in computer science and engineering, the electrical and electronics fields, or related fields. Your entry membership grade will be determined by your level of participation, contributions, education and/or experience in those fields.

Residence	Half-Year Mar 1-Aug 31	Full-Year Sept 1-Feb 29
United States	☐ \$41.00	☐ \$82.00
Canada	☐ \$38.50	☐ \$77.00
Europe, Africa, Middle East	☐ \$39.00	☐ \$78.00
Latin America	☐ \$33.50	☐ \$67.00
Asia, Pacific	☐ \$34.50	☐ \$69.00

*ACM members who join both IEEE and the Computer Society are entitled to a deduction of \$5 off the full-year rate; \$2.50 off the half-year rate.

TOTAL

(remittance in US dollars only)

☐ Check or Money Order
(payable to the IEEE)

☐ Visa

☐ Master Card

☐ American Express

Credit Card Number _____

Expiration Date _____

STUDENTS! ▶

You receive special rates and benefits. Call or send for a student application.

I hereby make application for Computer Society membership and, if elected, will be governed by IEEE's and the Society's Constitutions, Bylaws, and Statements of Policies and Procedures.

MAILING ADDRESS

Full signature _____

Date _____

First Name _____

Middle initial(s) _____

Last name _____

Street address _____

City _____

State/Country _____

Postal Code _____

Were you ever a member of the IEEE? ☐ Yes ☐ No Grade _____
If yes, please furnish

Membership No. _____

Date of birth _____

☐ Male ☐ Female

EDUCATION (highest level completed)

Course _____

Degrees received _____

Date _____

Name of educational institution: (Please print full name) _____

REFERENCE (an IEEE member; if unknown, a managerial person who knows you professionally)

OCCUPATION

Title or Position _____

Name (print in full) _____

IEEE Member No. (if applicable) _____

Firm Name _____

Position _____

Firm address _____

Street Address _____

City _____

State/Country _____

Postal Code _____

City _____

State/Country _____

Postal Code _____

Residents of Europe mail to:

Computer Society
of the IEEE
13, Avenue de l'Aiglon
B-1200 Brussels, Belgium
Telephone: 32-2-770-21-98

Payment to the Brussels office may be made by cheques in Belgian francs, British pounds sterling, German marks, Swiss francs, or US dollars, or by American Express, Eurocard, MasterCard, or Visa credit cards.

All others mail to:

Computer Society
of the IEEE
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-2578
Telephone: (714) 821-8380

Allow six to eight weeks
for delivery of publications.

PC0988

CALL FOR PAPERS

Call for papers and referees for *Computer*



Computer seeks articles for inclusion in upcoming special issues.

Autonomous Intelligent Machines is the theme scheduled for June 1989.

Suggested topics include

- distributed and parallel architectures for intelligent machines;
- multisensor integration for intelligent machine applications;
- real-time control and coordination of mobile robots and manipulators;
- machine intelligence and expert systems;
- robot programming;
- path planning and navigation; and
- learning and autonomous machines.

Papers must not have been previously published or currently submitted for publication elsewhere.

A 300-word abstract should be submitted as soon as possible. Eight copies of the full manuscript must be submitted by **Oct. 1, 1988**, to S. Sitharama Iyengar, Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803-4020, phone (504) 388-1495, E-mail: iyengar%lsu.edu@relay.csnet; or to Rangasami L. Kashyap, Department of Electrical Engineering, Purdue University, West Lafayette, IN 47907, phone

(317) 494-3473, csnet E-mail address: kashyap@ee.ecn.purdue.edu.

Authors will be notified of acceptance by **Jan. 10, 1989**. The final version of the manuscript is due no later than **Feb. 25, 1989**.

Persons interested in serving as referees are asked to send a note with a list of technical interests to Iyengar, Kashyap, or Bruce D. Shriver, *Computer* Editor-in-Chief, Department of Decision Sciences, University of Hawaii, 2404 Maile Way, Honolulu, HI 96822, E-mail address: shriv@uhccux.uhcc.hawaii.edu.

Visualization in Scientific Computing has been selected as the theme for August 1989. Tutorial, survey, descriptive, case-study, applications-oriented, or pedagogic manuscripts are sought.

Suggested topics include

- hardware strategies for scientific visualization; parallel architectures, supercomputing, workstations, and networks;
- users interfaces, paradigms and interactive techniques for visualization of 3D (and higher dimensional) models;
- scientific data analysis, manipulation, representation, and display

techniques, particularly volume visualization and methods for large multivariate data sets;

- dissemination of the results of interactive systems and animated images; standards; and televisualization and networks; and
- application of visualization techniques to science and engineering problems and data.

Papers must not have been previously published or currently submitted for publication elsewhere.

A 300-word abstract is due as soon as possible. Eight copies of the full manuscript must be submitted by **Dec. 1, 1988**, to Gregory N. Nielson, Computer Science Dept., Arizona State University, Tempe, AZ 85287-5406, phone (602) 965-2785, electronic mail address: nielson@enuxva.asu.edu.

Authors will be notified of acceptance by **Mar. 1, 1989**. The final version of the manuscript is due no later than **May 1, 1989**.

Persons interested in serving as referees are asked to send a note with a list of technical interests to Nielson or Bruce D. Shriver, *Computer* Editor-in-Chief, Department of Decision Sciences, University of Hawaii, 2404 Maile Way, Honolulu, HI 96822, E-mail address: shriv@uhccux.uhcc.hawaii.edu.



IEEE Transactions on Knowledge and Data Engineering will begin publication in March 1989. Papers are sought, particularly those emphasizing research, design, and development of knowledge and data-engineering methodologies, strategies, and systems. For information and submission requirements, contact C. V. Ramamoorthy, Computer Science Division, University of California, Berkeley, CA 94720, phone (415) 642-4751; or B. W. Wah, Coordinated Science Laboratory, University of Illinois, 1101 W. Springfield Ave., Urbana, IL 61801, phone (217) 333-3516.



11th International Conference on Software Engineering: May 15-18, 1989, Pittsburgh. Cosponsor: ACM. Submit paper by **Sept. 15, 1988**, to Richard Fairley, School

of Information Technology, George Mason University, Fairfax, VA 22030; or Dines Bjorner, Dansk Datamatik Center, Lundtoftevej 1 C, Lyngby DK-2800, Denmark.

Transactions on Systems, Man, and Cybernetics will publish a special issue on unmanned systems and vehicles. Submit abstract by **Sept. 15, 1988**, and full paper by **Nov. 1, 1988**, to Arun K. Sood, Dept. of Computer Science, George Mason University, 4400 University Dr., Fairfax, VA 22030-4400, phone (703) 323-2713.

ICCAL 89, Second International Conference on Computer-Assisted Learning: May 9-11, 1989, Dallas. Sponsor: Computer Learning Research Center. Submit extended abstract by **Sept. 15, 1988**, Hermann Maurer, IIG,

Schiesstattgasse 4a, A-8010 Graz, Austria, phone 43 (316) 70-255.



CAIA 89, Fifth IEEE Conference on Artificial Intelligence Applications: Mar. 6-10, 1989, Miami, Fla. Submit paper by **Sept. 20, 1988**, to Mark S. Fox or Roy A. Maxion, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, phone (412) 268-3832.



CHI 89, Conference on Human Factors in Computing Systems: Apr. 30-May 4, 1989, Austin, Texas. Cosponsor: ACM. Submit paper by **Sept. 26, 1988**, to Clayton Lewis, Dept. of Computer Science, Campus Box 430, University of Colorado, Boulder, CO 80309, phone (303) 492-6657.

SIGArt newsletter plans a special issue on knowledge acquisition in April 1989. Sponsor: ACM. Submit extended abstract or paper by **Sept. 26, 1988**, to Christopher Westphal, Knowledge Acquisition Material, BDM Corp., 7915 Jones Branch Dr., McLean, VA 22102, phone (703) 848-7910.

International Symposium on Database Systems for Advanced Applications: Apr. 10-12, 1989, Seoul, Korea. Sponsors: Information Processing Society of Japan, Korean Information Science Society. Submit paper by **Sept. 30, 1988**, to In-Sup Paik, Data Communications Corp. of Korea, 65-228, 3-Ga, Hangang-Ro, Yongoan-Ku, Seoul 140, Korea, phone 82 (02) 796-6606.

CAPE 89, Computer Application in Production Engineering Conference: Oct. 2-5, 1989, Tokyo. Sponsors: International Federation for Information Processing, Information Processing Society of Japan, Japan Society of Precision Engineering. Submit extended abstract by **Sept. 30, 1988**, to Business Center for Academic Societies of Japan, 2-40-14, Hongo, Bunkyo-ku, Tokyo 113, Japan, phone 81 (3) 817-5831.

Sixth Canadian Symposium on Instructional Technology: May 3-5, 1989, Halifax, Nova Scotia, Canada. Sponsor: National Research Council Canada. Submit abstract by **Sept. 30, 1988**, to Laurier Forget, Sixth Canadian Symposium on Instructional Technology, Conference Services, NRCC, Ottawa, Ontario K1A 0R6, Canada, phone (613) 993-9009.

 **Ninth International Conference on Distributed Computing Systems:** June 5-9, 1989, Newport Beach, Calif. Submit paper by **Oct. 1, 1988**, to Norman Schneidewind, Naval Postgraduate School, Code 545S, Monterey, CA 93943, phone (408) 646-2719.

CMC 89, Colorado Microelectronics Conference: Mar. 30-31, 1989, Colorado Springs, Colo. Sponsor: University of Colorado. Submit abstract by **Oct. 1, 1988**, to Technical Chair, CMC 89, Microelectronics Research Laboratories, University of Colorado, Box 7150, Colorado Springs, CO 90933-7150.

Multimedia 89, Second Comsoc International Multimedia Communications Workshop: Apr. 20-23, 1989, Ottawa, Canada. Sponsor: IEEE. Submit abstract by **Oct. 1, 1988**, to Nicolas D. Georganas, Multimedia 89, Faculty of Engineering, University of Ottawa, Ottawa, Ontario, Canada K1N 6N5, phone (613) 564-8222.

1989 International Conference on Robotics and Automation: May 14-19, 1989, Scottsdale, Ariz. Sponsor: IEEE. Submit paper by **Oct. 1, 1988**, to John M. Hollerbach, MIT Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139.

Eighth SIGAct-SOGMod-SIGArt Symposium on Principles of Database Systems: Mar. 29-31, 1989, Philadelphia. Sponsor: ACM. Submit abstract by **Oct. 10, 1988**, to Ashok K. Chandra, PODS, IBM T.J. Wat-

son Research Center, PO Box 218, Yorktown Heights, NY 10598, phone (914) 945-1752.

 **FTCS 19, International Symposium on Fault-Tolerant Computing:** June 21-23, 1989, Chicago. Submit abstract by **Oct. 14, 1988**, and final manuscript by **Nov. 21, 1988**, to S.M. Reddy, FTCS 19, Dept. of Electrical and Computer Engineering, University of Iowa, Iowa City, Iowa 52242, phone (319) 335-5196.

Fifth International Workshop on Software Specification and Design: May 19-20, 1989, Pittsburgh. Submit extended abstract or paper by **Oct. 15, 1988**, to Colin Potts, MCC, 9390 Research Blvd., Kaleido II Bldg., Austin, TX 78759, phone (512) 338-3629.

NECC 89, 10th National Educational Computing Conference: June 20-22, 1989, Boston. Submit paper by **Oct. 15, 1988**, to NECC 89, International Council for Computers in Education, University of Oregon, 1787 Agate St., Eugene, OR 97403-9905.

SETSS 90, Seventh International Conference on Software Engineering for Telecommunications Switching Systems: July 3-6, 1989, Bournemouth, England. Sponsor: Institution of Electrical Engineering. Submit synopsis by **Oct. 26, 1988**, to Conference Services, IEE, Savoy Pl., London WC2R 0BL, UK, phone 44 (1) 24-01-871.

Fourth Conference on Hypercube Concurrent Computers and Applications: Mar. 6-8, 1989, Monterey, Calif. Submit abstract by **Oct. 30, 1988**, to John Gustafson, Sandia National Laboratories, Division 3538, Bldg. MO-196, Albuquerque, NM 87185.

 **CHDL 89, Ninth International Symposium on Computer Hardware Description Languages and Their Applications:** June 19-21, 1989, Washington, DC. Cosponsors: International Federation of Information Processing, ACM. Submit paper by **Oct. 31, 1988**, to Franz Rammig, University of Paderborn, Warburger Str. 100, D4790 Paderborn, Federal Republic of Germany.

HCI International 89, Third International Conference on Human-Computer Interaction: Sept. 18-22, 1989, Boston. Submit abstract by **Oct. 31, 1988**, Gavriel Salvendy, School of Industrial Engineering, Grissom Hall, Purdue University, West Lafayette, IN 47907, phone (317) 494-5426.

Third International Conference on Foundations of Data Organization and Algorithms: June 21-23, 1989, Paris. Sponsor: INRIA. Submit paper by **Oct. 31, 1988**, to Witold Litwin, INRIA Rocquencourt, c/o Public Relations Dept., Domaine de Voluceau, 78153 Le Chesnay cedex, France, phone, 33 (1) 39-63-56-00.

Graphics Interface 89: June 19-23, 1989, London, Ontario, Canada. Sponsor: Canadian Man-Computer Communications Society. Submit paper by **Oct. 31, 1988**, to

Marceli Wein, Graphics Interface 89, National Research Council, Computer Graphics Section, Bldg. MC50, Montreal Rd., Ottawa, Ontario, Canada K1A 0R8.

Vision Interface 89: June 19-23, 1989, London, Ontario, Canada. Sponsor: Canadian Image Processing and Pattern Recognition Society. Submit paper by **Oct. 31, 1988**, to Terry Caelli, Vision Interface 89, Dept. of Psychology, Queen's University, Kingston, Ontario, Canada K7N 3N6.

 **IEEE Software:** July 1989. Articles are sought for a special edition on parallel-processing languages. Submit manuscript by **Nov. 1, 1988**, to Shreekanth Thakkar, Sequent Computer Systems, 15450 SW Koll Pkwy., Beaverton, OR 97006, phone (503) 627-9822.

 **IEEE Transactions on Computers** plans a special issue on distributed computing systems in August 1989. Submit manuscript by **Nov. 1, 1988**, to Steve S. Yau, Computer and Information Sciences Dept., University of Florida, Gainesville, FL 32611, phone (904) 335-8000.

 **19th International Symposium on Multiple-Valued Logic:** May 29-31, 1989, Guangzhou, China. Submit paper by **Nov. 1, 1988**, to J.C. Muzio, Dept. of Computer Science, University of Victoria, PO Box 1700, Victoria, B.C., Canada V8W 2Y2 (for the Americas); to C. Moraga, FB Informatik, Univ. of Dortmund, Postfach 500500, 4600 Dortmund 50, FRG (for Europe/Africa); or to Mou Hu, Shanghai Institute of Railway Technology, Shanghai, China (for Asia/Pacific).

Third International Conference on Image Processing: July 18-20, 1989, Warwick, England. Sponsor: Institution of Electrical Engineers. Submit paper by **Nov. 1, 1988**, to Conference Services, IEE, Savoy Pl., London WC2R 0BL, phone 44 (1) 24-01-871, ext. 222.

Congress 89, 11th World Computer Congress: Aug. 28-Sept. 1, 1989, San Francisco. Sponsor: International Federation for Information Processing. Submit paper by **Nov. 1, 1988**, to Herve Gallaire, EERC, Arabellstrasse 17, D-8000 Munich 81, Federal Republic of Germany, phone 49 (89) 92-69-91-00.

Sixth International Workshop on Database Machines: June 19-21, 1989, Deauville, France. Sponsors: INRIA, AFCET. Submit paper by **Nov. 1, 1988**, to Haran Boral, MCC, 3500 W. Balcones Center Dr., Austin, TX 78759 (in the US); or Pascal Faudemay, Labo. Masi., University of Paris 6, 4 Place Jussieu, 75252 Paris, Cedex 05, France (outside the US).

 **1989 IEEE Symposium on Research in Security and Privacy:** May 1-3, 1989, Oakland, Calif. Submit paper by **Nov. 2, 1988**, to Thomas A. Berson, Anagram Laboratories, 746 Forest Ave., Palo Alto, CA 94301, phone (415) 324-0100.

CALL FOR PAPERS

THE NEW IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING



THE COMPUTER SOCIETY



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

AIM

The new quarterly TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING aims to provide an international and interdisciplinary forum to publish results on the research, design, and development of knowledge and data engineering methodologies, strategies and systems. March 1989 will be the first issue.

TARGET AUDIENCE

Researchers, developers, managers, strategic planners, users, and others interested in state-of-the-art and state-of-the-practice activities in the knowledge and data engineering area

FOCUS

The key technical issues this Transactions will address include

- (a) acquiring and managing knowledge and data in developing and using information systems;
- (b) strategies to capture and store new knowledge and data;
- (c) methods to lessen the burden of software and hardware development and maintenance;
- (d) mechanisms to provide system modeling, design, access, and security and integrity control;
- (e) architectures, systems, and components to provide knowledge and data services within centralized and distributed information systems;

(f) designs to provide increased intelligence and ease of use through speech, voice, graphics, images and documents;

(g) techniques to provide improved overall functions and performance to meet new social needs; and

(h) the development of ways to prolong the useful life of knowledge and data and its graceful degradation.

SOME PERTINENT DATA AND KNOWLEDGE ENGINEERING TOPICS TO BE ADDRESSED

- Artificial Intelligence techniques
- Knowledge and data engineering tools and techniques
- Parallel and distributed processing
- Real-time knowledge bases and databases
- System architectures, integration and modeling
- Database design, modeling and management
- Query, design, and implementation languages
- Distributed database control
- Statistical databases
- Algorithms for data and knowledge management
- Performance evaluation of algorithms and systems
- Data communications aspects
- System applications and experience
- Knowledge-based and expert systems
- Integrity, security and fault tolerance

ARTICLE GUIDELINES

ARTICLE SELECTION PROCEDURES

This new periodical is at the Transactions level. The selection of articles for publication will follow the guidelines used by other IEEE Computer Society Transactions, such as the IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. A minimum of three reviews will be required for a decision to be made on each paper.

TYPE OF ARTICLES PUBLISHED

This Transactions will publish original results in research and development in areas relevant to knowledge and data engineering. Papers that can be submitted for consideration include those that have not previously been published in another journal, or are not currently being published, as well as those that have been published in Conference Proceedings, Digests, and Records and that have undergone substantial revision. Invited papers from leading authorities in the knowledge and data engineering area will also be published. Three types of papers will be published:

(1) Regular technical articles, which contribute to the understanding and advances in the knowledge and data engineering area (25-35 double spaced pages, including figures, tables, and references):

- (a) papers with extensive original results and
- (b) in-depth surveys;

(2) Concise short articles (maximum 12 pages): papers with results that are important and original and are presented in a concise form;

(3) Correspondence articles (maximum 3 pages): comments on previously published articles, short extensions to current results, critiques on previous results, responses from authors, and corrections to previously published articles. An effort will be made to shorten the turnaround time for concise papers and correspondence articles.

GUIDELINES FOR SUBMITTING PAPERS AND PROPOSALS ON SPECIAL ISSUES

(1) For invited papers and proposals for special issues, send 6 copies to:

C. V. Ramamoorthy, Editor-in-Chief
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
(415)642-4751, (415)642-1898
ram@ernie.berkeley.edu

(2) For all other submissions, send 6 copies of manuscript, complete with illustrations, abstract, and index terms, to:

Benjamin W. Wah, Associate Editor-in-Chief
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1101 West Springfield Avenue
Urbana, IL 61801
(217) 333-3516, (217) 244-7175
wah%aquinas@uxc.cso.uiuc.edu

IEEE copyright transfer form and general guidelines for submissions can be found in the January 1988 issue of IEEE TRANSACTIONS ON SOFTWARE ENGINEERING.

FURTHER INFORMATION

Any questions regarding the journal can be directed to either listed Editor. For complete submission guidelines, circle the reader service card at the back of this magazine.

Reader Service Number 5

BOOK REVIEWS

Editor: Wiley McKinzie, School of Computer Science and Technology, Rochester Institute of Technology, Rochester, NY 14623; Compmail, w.mckinzie; CSnet, wrm@rit

Programming Windows

Charles Petzold (Microsoft Press, Redmond, Wash., 1988, 852 pp., \$24.95)

This well-written tutorial text is strictly for experienced IBM PC programmers who need to learn how to make their programs run under Microsoft Windows with DOS. It will not help the everyday PC user working with, or planning to work with, the Windows operating environment. Its sheer size will probably deter anyone who doesn't have a real need to write code to run with Windows.

In spite of its excellence, the book by itself has limited utility. As Petzold observes, the *Windows Programmers Reference* must always be close at hand. In addition to Windows 2.0 (or Windows/386), programming for Windows actually requires the Windows Software Development Kit 2.0 and the Microsoft C compiler 4.0 or 5.0 (or equivalent with Windows capability). Working with dynamic link libraries (covered in Chapter 18) also requires a macro assembler.

Although Windows can run on a plain 640-Kbyte PC with graphics display (preferably EGA or VGA), it is frustratingly slow. The author's 8-MHz AT, with EGA, 1.5-Mbyte above board, and mouse should probably be viewed as a minimum useful system. Windows consumes lots of memory as well as lots of machine cycles! Compounding the complexity is the usual need to provide full functionality from the keyboard when the mouse is sick.

The author exhibits good sensitivity to the needs of the reader/student. He provides good rationales, not found elsewhere, for many of the conventions and seeming quirks of Windows. He also shows many examples of why a traditional or apparently obvious approach produces problems. He assumes the reader will understand the profusely

used C programming examples, but does not expect assembly language proficiency. In fact, he expends perhaps a bit too much effort explaining the nuances of the Intel 8086 segmented architecture.

As the first book of its kind it may become the definitive technical work on Windows.

The book is authoritative. It's published by Microsoft, the creator of Windows, and the author had access to Microsoft staff during the book's preparation. Although *Programming Windows* is his first book, Petzold is well respected for his many technical articles in PC-related periodicals.

In a sense the book introduces a new world of large-system multitasking on PCs. The entire book really deals with the substantial additional overhead required to make your flawless DOS program work in the Windows environment—an environment in which Windows owns most of the system resources and your application must negotiate for their use in contention with whatever other applications the user has initiated. In particular, the user, not your application program, has primary control of the display screen. Whether user input—by either mouse or keyboard—belongs to your application depends in part on the cursor's screen location. Windows is in charge: It will make unexpected demands on your application, tell you only what it decides you need to know, and execute your requests only when (or if) it

decides the time is right.

The formidable task of learning Windows doesn't permit incremental accomplishment. You can't just do simple things; you must do everything the Windows environment demands or you can't play. Your application program, no matter how big or important, will appear to be just one cog in Windows' big machinery. If the user elects to shrink your window to a tiny square in the corner, your program must do its business in some acceptable manner from that square. The book does a great job of developing and selling these perspectives.

The author has organized the book very logically. The 18 chapters are divided among five sections: Basic Concepts, Getting Input, Using Resources, The Graphic Device Interface, and Data Exchange and Links. A 10-page, three-column, small-print index provides important detailed access to the text. Nevertheless, the book's lack of appendices that gather command lists or calls into ready reference format limits its use as a mainstream reference. You must refer to each section in turn to learn how to accomplish the whole family of tasks required by Windows.

The book is laced with detailed—and progressively more complex—examples, mostly written in C. A shaded background makes the valid examples stand out nicely. Incomplete code fragments and bad examples (what not to do) are not highlighted and blend quietly into the text.

Another nice feature is the use of so-called Hungarian notation for variable names. This prefixes the name with one to four lowercase characters identifying the data type and, where needed, the pointer type. Variable names in the examples use upper- and lowercase wisely and are as long as necessary to make their meaning clear and unique. (Windows' built-in function names are

similarly well structured and up to 26 characters long.)

A set of four companion disks (\$29.95 from the publisher) is a worthwhile adjunct. The fully loaded disks are organized with each chapter's files in a separate subdirectory. They contain all of the book's examples (except for the "how-not-to-do-it" code fragments), including several useful utilities. In addition to the source as listed in the text, most examples include a related definition file and a compiled executable file.

Windows presents a dramatically different environment for the PC programmer. It takes over nearly all of the functions that DOS previously provided (though it may make its own use of DOS's interrupts). Although it intercepts Interrupt 21H DOS calls, Windows itself is not interrupt-driven. Instead it uses far calls for communication. This generates some new perspectives. Essentially, these calls post messages that must be polled regularly to see what's new and decide possible action. Since these calls are made among program modules that load (and unload) at various times, unresolved references must be dynamically linked each time a module loads. The Windows Software Development Kit includes a new linker that creates a "New Executable" exe file format to accommodate the needed references for dynamic linking.

Windows' array of about 450 function calls suggest both the power of this environment and the difficulty of learning to operate in it. The book does an excellent job of showing how these functions are used and why so many exist. The number of calls also suggests why this is such a huge book—the author must cover a lot of material, which he does with great care, completeness, and sympathy for the reader.

The book has many "forward references"—situations where parts of the sample code and its explanation are covered in later chapters. Explaining something as complexly interactive and seemingly convoluted as Windows makes this unavoidable. Fortunately, the references are always specific, so the reader can skip around efficiently.

In summary, this is an authoritative, well-written, tutorial-type book, but only for experienced programmers with an adequate C background to handle the examples. I commend the author for a difficult job well done. As the first book of its kind it may become the definitive technical work on Windows.

Charles B. Stott
Charles B. Stott Associates

MAP/TOP Networking: A Foundation for Computer-Integrated Manufacturing

Vincent C. Jones (McGraw-Hill, New York, 1988, 289 pp., \$54.50)

This book provides a detailed overview of the MAP (Manufacturing Automation Protocol) and TOP (Technical and Office Protocol) networks. The book's technical content is up to date, and the fact that MAP and TOP networks are among the primary local area networks being introduced for factory and office automation makes its publication very timely.

The author puts each piece in its proper place by describing the background behind almost every portion of the protocols. Although the book's intended audience is computer scientists and engineers who are designing computer-integrated manufacturing applications, it should also meet the needs of managers and students. For example, the preface contains detailed

carefully describes why computer networks exist and what services they provide.

A very simple example in Chapter 3 demonstrates the need for protocols. The author then devotes considerable space to MAP/TOP protocols and standards, explaining the need for protocol layers and examining each one. The author makes every effort to systematically build up the reader's knowledge base. Chapter 4, titled "The Genesis of MAP/TOP," begins with a very clear discussion of the need for standards and how MAP and TOP began. The author also explains why MAP 2.1 and TOP 1.0 are such significant milestones in MAP/TOP development.

After this the real detail starts. Four chapters explain the specific features of MAP and TOP. The author details the differences between MAP 2.X and MAP 3.0, providing many helpful hints to make upgrading a lot easier when MAP 3.0 components become available.

Three chapters then lead the reader through defining and developing network requirements, selecting the proper network for a given application, and finally installing and managing the network. The author offers a number of examples and practical hints, and then concludes with an examination of current networking trends.

The frequent use of acronyms that were new to me posed my only problem with the book. The author explains each acronym again as he uses it in a new chapter, but it did slow my reading considerably. All in all, though, I am very pleased with the book and recommend it to anyone interested in MAP/TOP networking.

Robert C. Stanley
IBM Boca Raton

MAP and TOP networks are among the primary local area networks being introduced for factory and office automation.

aids for managers who might wish to skim through some of the detailed technical sections and skip others completely.

Chapter 1 uses two real-world examples to detail the benefits of MAP/TOP. Chapter 2 offers a good introduction to computer networks and establishes some common terminology for the rest of the book. The author

Reviewers wanted

If you are interested in reviewing books for *Computer*, please submit your name, address, and a list of areas of interest and expertise to the Book Reviews Editor at the address below. Publishers should submit recent books for review consideration to the same address.

Wiley R. McKinzie
Office of the Dean
School of Computer Science and Technology
Rochester Institute of Technology
One Lomb Memorial Drive
PO Box 9887
Rochester, NY 14623

68000 Microcomputer Systems: Designing and Troubleshooting

Alan D. Wilcox (Prentice Hall, Englewood Cliffs, N.J., 1987, 579 pp., \$42.95)

A superior class of computer textbooks has arisen during the past five years. These books characteristically are designed as a total teaching environment and have been tested in the classroom. They combine a pedagogical soundness with complete technical details and a dedication to the theoretical fundamentals of the topic. Comer's books on operating systems exemplify this class. Wilcox's contribution to computer science and engineering education is a welcome addition to this elite group and may well be the best of this class.

The book offers a complete tutorial for the design and construction of a fully functional 68000-based microcomputer. It supplies all of the information needed to initiate, plan, implement, test, and use the system. In the process, it establishes a set of principles for the effective and trouble-free design of such systems in general, and provides a solid foundation in the theory and practice of computer engineering. For this alone, the book should be a mandatory acquisition for any practicing or prospective hardware design engineer.

The style and format of the preface and introduction establish the book's high presentation quality. The author describes the content, intended audience, and suggested uses of the text, and supplies sources for additional books, test equipment, software, and hardware. A concise chapter-by-chapter summary in the introduction offers the best overview of the contents.

Each chapter has a consistent format, beginning with a discussion of how the chapter's contents fit into the overall project, and concluding with a summary, an excellent set of exercises, and a list of additional relevant literature.

The early chapters introduce the design process and project-planning format to be used throughout. The author makes excellent use of lists, flowcharts, and font changes for emphasis. He explains each phase of the design process, justifies it, and links it to its neighbors with a smooth, conversational writing style. The author has the rare ability to supply exactly the right amount of detail in his explanations.

The third chapter's potpourri of topics provides the book's first look at hardware. The author describes TTL- and CMOS-based designs and the oper-

ating conditions and electrical characteristics of some of the chips. He also presents sample circuit designs and related calculations. The chapter contains a superb discussion of clock circuits and clock timing, a list of dos and don'ts for electrical noise, and a brief section on software design.

Chapter 4 brings us back to design principles with a detailed description of the format for a technical manual, with one student's effort included as an appendix. The inclusion of student work throughout the text is a good feature; the outstanding work will motivate and serve as a model for others.

Wilcox has produced a polished, complete, and very valuable volume.

Chapter 5 contains a case study for the design and implementation of a system clock. The study rigorously adheres to the design models introduced earlier, and detailed functional block, circuit, and timing diagrams accompany the design. The chapter also offers step-by-step instructions for constructing a prototype clock.

At this point, the reader should be ready to tackle the 68000 system. A two-page project plan and time schedule establishes the foundation for the implementation, as well as for most of the remainder of the book. The author describes the register set and the major classes of input/output signals. The explanation of the 68000 bus operation includes detailed instruction decoding and timing diagrams and a thorough treatment of read and write cycle operation.

Having whetted the reader's appetite for some hands-on lab work, the author moves away from the central theme to describe the equipment needed to test the system, including an LED, volt-ohm meter, logic probe, oscilloscope, in-circuit emulator, and logic analyzer. The author describes a working 68000 system—the Motorola Educational Computer Board—and gives a brief series of tests using the above equipment. To make the most effective use of the information in this chapter, instructors should design a more detailed lab exercise for this equipment.

The reader now should be ready to put a basic 68000 system together, using an S-100 (IEEE Std-696) prototype board and the circuit diagrams supplied to build a power supply, system clock, and reset circuit, and then combining them with the microprocessor. The processor can now function in a free-running mode.

The goal is to run simple programs, so the addition of some memory is the next logical step. The author presents the memory design with the same expertise, detail, and completeness that pervades the entire book. The author describes specifications and circuits for RAM and EPROM, lays out the components on the S-100 CPU board, and tests the memory using the Motorola Tutor monitor.

The remaining modules complete the system. The author describes all software and hardware aspects of the Motorola 6850, which is used as the I/O processor because of its compatibility with Tutor. The chapter covering the interrupt logic module contains one of the best discussions available on error processing for the 68000. Finally, addition of a bus interface module allows the system to access additional memory and other I/O devices.

The project development concludes with a discussion of the Tutor firmware and cross-assembly techniques for producing downloadable 68000 code. Because Tutor has limited functionality, the author provides information to bring up the disk-operating system CP/M-68K.

Students must have a strong background in fundamental electrical principles and circuit design before tackling this project. Furthermore, an instructor must understand the overall project as thoroughly as the author does before beginning the course. Successful management of the considerable detail supplied and student construction of each module and the integrated system requires an expert guide through the technical maze.

I am impressed with all aspects of this book. I was unable to find any defects in content, style, or organization. Wilcox has produced a polished, complete, and very valuable volume. It should become a standard text for upper-division and graduate computer engineering students and is a bible of sound design techniques for the practicing computer engineer.

John D. Holt
California State University, Fresno

Compact Disc-Interactive: A Designer's Overview

Philips International (McGraw-Hill, New York, 1988, 239 pp., \$39.95)

A normal part of product planning and development in high-technology companies consists of preparing a variety of documents specifying functional objectives, likely market, product life-cycle, performance specifications, technical specifications, manufacturing requirements, etc. Countless departments write, review, and revise these documents countless times as a new product moves from concept to first customer shipment, and even afterward as the product moves through its life-cycle.

This book appears to be a collection of such documents relating to the products embraced by the term "compact disc-interactive," established as an international standard in March 1987. As such, the book is absolutely terrible and should be bypassed by anyone interested in this technology.

CD-I capitalizes on the 500-700 Mbyte storage capacity of laser optical compact discs to code (in read-only format) digital data, including computer programs, text, audio, computer graphics, and captured digitized still images. CD-I's ability to store digitized audio or picture data in various resolutions allows it to play back lower-quality audio (when it captures a small amount of data at a slow sampling rate) or high-quality stereo (when it captures a large amount of data at a faster sampling rate).

A key element in the playback of CD-I functions is a computer-based consumer workstation product that will bring together the data on the disc and provide high-level special effects and interactivity. The workstation will be compatible with existing consumer-level television and audio equipment, including existing compact audio discs.

Another important component of CD-I products is the authoring methodology and hardware/software systems used to develop applications using CD-I special effects. In many ways, the requirements for these authoring systems resemble those for highly interactive computer-controlled laserdisc applications.

As "A Designer's Overview" this book falls far short of introducing new information for those readers likely to become CD-I designers. Except for current CD-I proselytizers, most potential CD-I application designers probably have already designed interactive laserdisc applications. This book contains

nothing new for them, except the message that developers will have to know CD-I system performance specs to optimize data presentation for consumer users.

In spite of all these warnings, the book does not specifically discuss the authoring workstation in which all this optimizing must be done. Any authoring system that does not automatically

There is a great deal of redundancy and the subject matter is generally too vague to be useful to serious designers.

attend to such matters, and expects the designer to worry about these issues at the expense of the creative process, is simply a lousy authoring system.

This book was edited by the Philips International technical staff, who developed the technology with Sony. Let's hope these people do a better job at

product development than they did editing the book. The text is riddled with typographical and spelling errors, erratic writing style, and even poor physical layout.

The book contains a great deal of redundancy, and its treatment of the subject matter is generally too inconsistent and vague to be of much use to serious designers. In addition, the examples appear to be experiences acquired while developing the prototypical applications needed to demonstrate the products when they become available.

That, too, is a problem. After all, none of the CD-I products discussed in this book are generally available. Consequently, no body of knowledge or experience yet exists from which to draw a useful designer's overview. The real experiences already developed in the 10-year-old interactive laserdisc industry are much more useful than the few gushing examples cited in this text. If the folks at Philips ignore that body of knowledge, they are doing themselves and eventual CD-I authors a potentially disastrous disservice.

Don't judge CD-I by this book. Laser optical disc technology and even CD-I is worth far more than this book indicates; the book itself isn't worth the paper it was printed on.

Sorel Reisman
California State University, Fullerton

Law and the Software Marketer

Frederick L. Cooper III (Prentice Hall, Englewood Cliffs, N.J., 1988, 319 pp., \$39)

The first two parts of this book provide one of the best and most current discussions of software protection and the limitation of exposure to legal liability I have seen. Part III provides a very practical game plan for the protection of software rights. Because the author's background focuses on both the computer and communications fields, the book is extremely current and relevant to the rapid changes in software technology. The author calls the primary area of focus "limited distribution/application software."

The author assumes the reader has no background in this area and carefully defines intellectual property, software, trade secrets, copyright, public domain, registration, works for hire, infringe-

ment, and remedies. He also provides several relevant "how-to" appendixes.

The book offers an excellent introduction to newly developing concerns such as LANs, mask works, exclusive rights, performance assurances, and international protection and liability.

The material is readable for the novice (it uses a football game plan that is somewhat trite), but at the same time contains much technical information of unusual relevance and interest to experienced professionals.

All in all, *Law and the Software Marketer* will be a classic desk reference and resource. It is an excellent update of earlier books, such as Michael Gemignani's *Law and the Computer* (CBI Publishing, 1981). I recommend it for anyone concerned with these topics.

R. Waldo Roth
Taylor University

NEW LITERATURE

Teleteaching proceedings. Elsevier Science Publishing has announced the release of *Remote Education and Informatics: Teleteaching* (ISBN 0-444-70418-3, 254 pp., \$65.75), comprised of the proceedings of an IFIP TC3 conference in Budapest, Hungary, in 1986. The conference's purpose was to prove that "distance education" is the only possible way of teaching informatics to adults. Topics include the use of CD-ROM, television, home computers, interactive optical disks, and expert systems.

In the US, contact Elsevier Science Publishing, P.O. Box 1663, Grand Central Station, New York 10163. Elsewhere, contact Elsevier Science Publishing, P.O. Box 211, 1000 AE Amsterdam, The Netherlands.

Environmental implications. The World Resources Institute has published *The Shrinking Planet: US Information Technology and Sustainable Development* (ISBN 0-915825-20-1, 88

pp., \$10, paperback) by John Elkington and Jonathan Shopley. The authors examine the environmental implications and applications of current information and communication technologies. The book also reviews emerging trends in information technology and some of the attendant issues for policy makers, particularly those relating to economic growth in developing countries.

Contact The World Resources Institute, 1750 New York Ave., N.W., Washington, D.C. 20006.

Storing images. The cross-breeding of computers, laser disks, and microfilm is revolutionizing the storage and retrieval of images, according to a new Frost & Sullivan study, *Analog and Digital Micro-imaging in Europe* (No. E1009, 236 pp., \$2,750). The report predicts that laser disk image storage systems will begin to make a significant impact in 1988, showing a 40-percent gain over 1987.

In the US, contact Customer Service,

Frost & Sullivan, 106 Fulton St., New York 10038. Elsewhere, contact Customer Service, Frost & Sullivan, Sullivan House, 4 Grosvenor Gardens, London SW1W 0DH.

Business software directory. The International Database Management Association has released the seventh edition of *The Business Software Catalog*, containing descriptions of the top 1,500 business and government software packages in 200 application software classifications cross-referenced by producing company, vertical market classification, and product name. All software listed is SMA industry standard. A free brochure listing all vertical market classifications is available upon request.

For more information, contact the International Database Management Association, 10675 Treena St., Suite 103, San Diego, CA 92131; phone (619) 578-3152.



Have you heard about our...

Technical Committee on Multiple-Valued Logic?

For information on this, or any of our 32 other TCs, members* may contact

THE COMPUTER SOCIETY
10662 Los Vaqueros Circle
Los Alamitos, CA 90720
Telephone: (714) 821-8380

*Nonmembers: Join us. For membership information, circle number 202 on the reader service card.

NOW LEASING

OFFICE, WAREHOUSE, DISTRIBUTION,

R&D. IMMEDIATE OCCUPANCY.

LOADING DOCKS, DRIVE-INS, FREE

PARKING, QUICK ACCESS TO I-95,

I-395, NATIONAL AIRPORT & DC.

Call now

PLAZA 500

For full details contact Julie Donatelli
D&K Management, Inc.
Artery Plaza—Suite 310
7200 Wisconsin Ave.,
Bethesda, MD 20814
(301) 654-0700

ADVERTISER INDEX

Aerospace Computer Security Applications Conference	10-11
Amdahl Corporation	113
CACI Products Company	1
Computer Society Membership	105, 119, 127
Computer Society Ombudsman	9
D&K Management, Inc.	127
Databases in Parallel and Distributed Systems	66-67
Design Automation Conference	107
Entity-Relationship Approach Conference	89
Executive Audial Rehabilitation Society	4-5
Fault-Tolerant Computing Symposium	57
The Foxboro Company	42
Gordon Bell Awards	9
IEEE Globecom '88 Workshop	117
IEEE Transactions on Knowledge and Data Engineering	122
Interactive Development Environments	Cover II
Meta Software	29
Multiple-Valued Logic Symposium	56
National Center for Excellence in Metalworking Technology	31
Network Management and Control Workshop	71
NeuralWare Inc.	Cover IV
Software for Strategic Systems Conference	75
University of California, Los Alamos	65
Visible Systems Corporation	12
Classified Advertising	108-111

FOR DISPLAY ADVERTISING INFORMATION, CONTACT:

Northern California and Pacific Northwest: Roy McDonald Assoc. Inc., 5915 Hollis St., Emeryville, CA 94608; (415) 653-2122.
 Jim Olsen, P.O. Box 696, Hillsboro, OR 97123; (503) 640-2011.
Southern California and Mountain States: Richard C. Faust Co., 24050 Madison St., Suite 100, Torrance, CA 90505; (213) 373-9604.
Southwest: The House Co., 3000 Wesleyan, Suite 345, Houston, TX 77027; (713) 622-2868.
Midwest: Robert Acker & Assoc., 480 Central Ave., Northfield, IL 60093; (312) 441-6050.
East Coast: Atlantic Representative Group, 349 Maple Place, Keyport, NJ 07735; (201) 739-1444.
New England: Arpin Associates, PO Box 227, Weston, MA 02193; (617) 899-5613.
 Paul Gillespie, 40 Sterling St., Somerville, MA 02144; (617) 625-1777.
Europe: Heinz J. Gorgens, Parkstrasse 8a, D-4054 Nettetal 1 - Hinsbeck (F.R.G.); phone: (0 21 53) 8 99 88; telex 841(17)2153310=HJG tlx d.

Advertising Director: Dawn Peck. For production information, conference, and classified advertising, contact Heidi Rex or Marian Tibayan.

COMPUTER, 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; phone (714) 821-8380; Fax (714) 821-4010.

PRODUCT INDEX

	RS#	Page#
BOARDS		
Coprocessor board	36	99
Microsystems announcement	135-147	106
CIRCUITS		
IC announcements	120-132	104
CONFERENCES & COURSES		
Aerospace Computer Security	—	10-11
Databases in Parallel and Distributed Systems	—	66-67
Design Automation	—	107
Entity-Relationship Approach	—	89
Fault-Tolerant Computing	—	57
Multiple-Valued Logic	—	56
Network Management and Control	—	71
Network Operations and Mgmt.	—	117
Software for Strategic Systems	—	75
DATA COMMUNICATIONS		
Gateway product	49-50	101
Neural architecture	38	99
MEMORY/STORAGE EQUIPMENT		
DOS	42	100
OTHER PRODUCTS & SERVICES		
Design competition	2	4-5
Gordon Bell awards	—	9
Real Estate	6	127
PUBLISHERS		
CS membership	—	105, 119, 127
CS ombudsman	—	9
Knowledge and Data Engineering	5	122
RECRUITMENT		
Classified advertising	—	108-111
Technical professional	—	31, 42, 65, 113
SOFTWARE		
Ada development environment	37	99
Ada tool	20-25, 51	91, 93, 95, 96, 97, 102
CAE	53-56	102
CASE tool	3	12
Expert system software	41	100
Fortran compiler	48	101
Mathematical package	52	102
Neural design software	58	102
Picture software engineering	57	102
Software development	1	C.II
SYSTEMS		
Computer	30, 32, 39	98, 100
Computer-room system	34-35	99
Database management system	46	101
Design tool	4, 44	29, 100
Electromagnetic signal	61	103
Expert system development	31, 45, 63	98, 101, 103
Graphics tool	40	100
Knowledgebase management	47	101
Neural development tool	7	C.IV
OEM systems	64	103
OS/2	43	100
Processing application	62	103
Simulation package	—	1
Workstation	33, 59-60	98, 103

COMPUTER

September 1988 issue (card void after February 1989)

Name _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____

Please send (circle those you want)

- | | |
|----------------------------|---------------------------------------|
| 201 Publications catalog | 203 Student membership information |
| 202 Membership information | 204 Application for senior membership |

Reader interest

(circle what you liked;
add comments on the back)

- A Graphics Supercomputer
- B Soviet Computers
- C Fuzzy Concepts
- O R&D Management
- E President's Message
- F Standards
- G Computer Society News
- H CS Election
- J Update
- K New Product Reviews
- L New Products
- M IC Announcements
- N Microsystems
- P Conferences
- Q Calendar
- R Call for Papers
- S Book Reviews
- T New Literature

Product inquiries

(circle numbers for products and advertisements
on which you want more information)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

1

COMPUTER

September 1988 issue (card void after February 1989)

Name _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____

Please send (circle those you want)

- | | |
|----------------------------|---------------------------------------|
| 201 Publications catalog | 203 Student membership information |
| 202 Membership information | 204 Application for senior membership |

Reader interest

(circle what you liked;
add comments on the back)

- A Graphics Supercomputer
- B Soviet Computers
- C Fuzzy Concepts
- O R&D Management
- E President's Message
- F Standards
- G Computer Society News
- H CS Election
- J Update
- K New Product Reviews
- L New Products
- M IC Announcements
- N Microsystems
- P Conferences
- Q Calendar
- R Call for Papers
- S Book Reviews
- T New Literature

Product inquiries

(circle numbers for products and advertisements
on which you want more information)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

2

COMPUTER

September 1988 issue (card void after February 1989)

Name _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____

Please send (circle those you want)

- | | |
|----------------------------|---------------------------------------|
| 201 Publications catalog | 203 Student membership information |
| 202 Membership information | 204 Application for senior membership |

Reader interest

(circle what you liked;
add comments on the back)

- A Graphics Supercomputer
- B Soviet Computers
- C Fuzzy Concepts
- O R&D Management
- E President's Message
- F Standards
- G Computer Society News
- H CS Election
- J Update
- K New Product Reviews
- L New Products
- M IC Announcements
- N Microsystems
- P Conferences
- Q Calendar
- R Call for Papers
- S Book Reviews
- T New Literature

Product inquiries

(circle numbers for products and advertisements
on which you want more information)

1	21	41	61	81	101	121	141	161	181
2	22	42	62	82	102	122	142	162	182
3	23	43	63	83	103	123	143	163	183
4	24	44	64	84	104	124	144	164	184
5	25	45	65	85	105	125	145	165	185
6	26	46	66	86	106	126	146	166	186
7	27	47	67	87	107	127	147	167	187
8	28	48	68	88	108	128	148	168	188
9	29	49	69	89	109	129	149	169	189
10	30	50	70	90	110	130	150	170	190
11	31	51	71	91	111	131	151	171	191
12	32	52	72	92	112	132	152	172	192
13	33	53	73	93	113	133	153	173	193
14	34	54	74	94	114	134	154	174	194
15	35	55	75	95	115	135	155	175	195
16	36	56	76	96	116	136	156	176	196
17	37	57	77	97	117	137	157	177	197
18	38	58	78	98	118	138	158	178	198
19	39	59	79	99	119	139	159	179	199
20	40	60	80	100	120	140	160	180	200

3

Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

**Computer
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA**

Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

**Computer
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA**

Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

For reader service inquiries, use other side

PO box address for
reader service cards only

PLACE
STAMP
HERE

**Computer
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA**



THE COMPUTER SOCIETY

A member society of the Institute of Electrical and Electronics Engineers, Inc.

Executive Committee

President: Edward A. Parrish, Jr.*
Vanderbilt University
School of Engineering
Nashville, TN 37235
(615) 322-2762

President-Elect: Kenneth R. Anderson*

Vice Presidents

Standards: Helen M. Wood (1st VP)*

Conferences and Tutorials: Joseph E. Urban (2nd VP)*

Area Activities: Willis King†

Computer Society Press: Duncan H. Lawrie

Education: Gerald L. Engel†

Membership and Information: Merlin G. Smith†

Publications: James H. Aylor†

Technical Activities: Laurel V. Kaleda

Secretary: Duncan H. Lawrie

Treasurer: Barry W. Johnson†

Past President: Roy L. Russo*

Division V Director: Harriett Rigas†

Division VIII Director: H. Troy Nagle, Jr.†

Executive Director: T. Michael Elliott†

*voting member of the Board of Governors
†nonvoting member of the Board of Governors

Board of Governors

Term Ending 1988

Mario Barbacci
Victor Basili
Paul Borrill
Lorraine M. Duvall
Michael Evangelist
Allen L. Hankinson
Laurel V. Kaleda
Ted Lewis
Ming T. Liu
Earl E. Swartzlander, Jr.

Term Ending 1989

Bill D. Carroll
Lansing Hatfield
Duncan H. Lawrie
David Pessel
Susan L. Rosenbaum
Sallie V. Sheppard
Bruce D. Shriver
Harold S. Stone
Akihiko Yamada
Marshall C. Yovits

Next Board Meeting

November 18, 1988—8:30 a.m.
Hyatt Orlando, Kissimmee, FL

Senior Staff

Executive Director: T. Michael Elliott
Editor and Publisher: True Seaborn

Director, Computer Society Press: Chip G. Stockton

Director, Conferences: Anne Marie Kelly

Computer Society Offices

Headquarters Office

1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Phone: (202) 371-0101

Telex: 7108250437 IEEE COMPSO

Publications Office

10662 Los Vaqueros Circle
Los Alamitos, CA 90720

Membership and General Information: (714) 821-8380

Publications Orders: (800) 272-6657

European Office

13, Avenue de l'Aquilon
B-1200 Brussels, Belgium
Phone: 32 (2) 770-21-98
Telex: 25387 AVVALB

Asian Office

Ooshima Building
2-19-1 Minami-Aoyama, Minato-ku
Tokyo 107, Japan

Use the Reader Service Card to obtain the following information:

- Membership application—student #203, others #202
- Periodicals subscription form for individuals #200
- Periodicals subscription form for organizations #199
- Publications catalog #201
- Standards working groups list #195
- COMPMail+ international electronic mail/database brochure #194
- Technical committee list/application #197
- Chapters lists, start-up procedures—student/regular #193
- Student scholarship information #192
- Awards description/nomination forms #198
- Volunteer leaders/staff directory #196
- IEEE senior member application #204

Purpose

The Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 90,000 members worldwide, and provides a wide range of services to members and nonmembers.

Membership

Members receive the acclaimed monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others seriously interested in the computer field.

Publications and Activities

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field; plus news, conferences, calendar, interviews, new products, etc.

Periodicals. The society publishes six magazines and three research transactions. Refer to membership application or request information as noted above.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 100 titles every year.

Standards Working Groups. Over 100 of these groups produce IEEE standards used throughout the industrial world.

Technical Committees. Over 30 TCs provide newsletters, interaction with peers in specialty areas, and have direct influence on standards, conferences, education, etc.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing sciences accreditation.

Chapters. Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

European Office

Payments for Computer Society membership and publication orders are accepted by cheques in Belgian, British, German, Swiss, or U.S. currency, or by American Express, Eurocard, MasterCard, or Visa credit cards.

Ombudsman

Members experiencing problems—magazine delivery, membership status, unresolved complaints—may write to the ombudsman at the Publications Office.

Neural Network Development Tools

on the IBM PC, XT, AT and SUN workstations
for

**rapid prototyping and concept testing
of neural network designs**

from

NeuralWare™

NeuralWorks Professional II

Includes 13 network types plus the ability to define your own network, 14 learning rules, 10 transfer functions, 11 summation functions.
IBM PC \$995; SUN \$2,995.

NeuralWorks Explorer

Lets you get your feet wet in neural computing without investing a lot of money.
IBM PC \$199; SUN \$795.

Seminars

Five-day, hands-on, applications oriented. Available throughout the country. Also available for customized in-house presentation.

Custom engineering

Let us help solve your special problem.

Call TODAY or write for information about NeuralWare's software, seminars, and custom engineering services. Ask for Jane Klimasauskas, Vice-President Sales & Marketing.

Pop-up menus make NeuralWorks Professional II easy to learn, easy to use

Full color screens and effective graphics guide you through the network-building and testing process

Documentation includes extensive introduction to get you up to speed in neural computing

Solve modeling and forecasting problems

- finance and economics
- servo control
- sensor processing
- CAD/CAM modeling

Solve signal processing problems

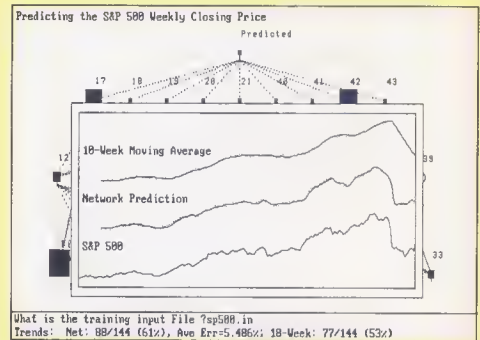
- noise filtering
- matched filters & speech recog
- data compression

Solve expert systems problems

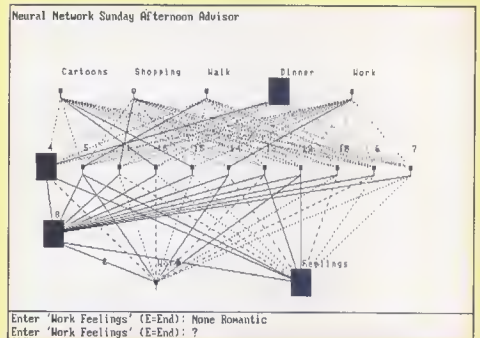
- adaptive expert systems

NeuralWare, Inc.

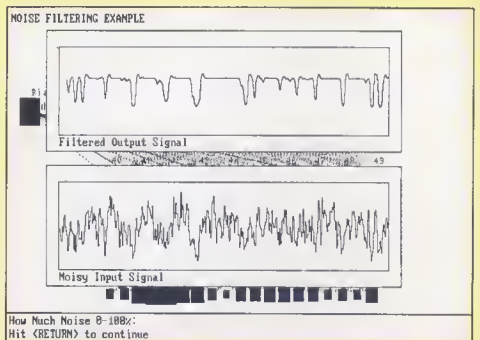
103 Buckskin Court
Sewickley, PA 15143 U.S.A.
412-741-5959



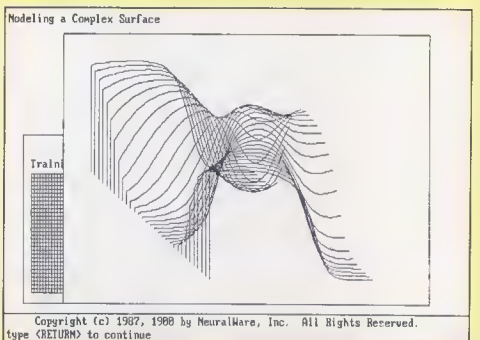
Stock Market Forecasting



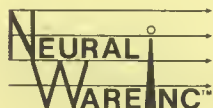
Expert Systems



Noise Filtering



Modeling



NeuralWare™ is applied neural computing.